DOCUMENT RESUME

ED 196 399 IR 008 840

AUTHOR Keeler, F. Laurence
TITLE System for Computer Automated Typesetting (SCAT) of Computer Authored Texts.
INSTITUTION Chief of Naval Education and Training, Orlando, Fla. Training Analysis and Evaluation Group.
REPORT NO TAEG-R-88
PUB DATE Jul 80
NOTE 153p.

EDRS PRICE MF01/PC07 Plus Postage.
DESCRIPTORS *Computer Graphics; *Computer Oriented Programs; Evaluation; Flow Charts; *Input Output Devices; *Printing; Programed Instructional Materials
IDENTIFIERS *System for Computer Automated Typesetting

ABSTRACT

This description of the System for Automated Typesetting (SCAT), an automated system for typesetting text and inserting special graphic symbols in programmed instructional materials created by the computer aided authoring system AUTHOR, provides an outline of the design architecture of the system and an overview including the component requirements, a flow chart and examples of system input data, composed text stream, and typeset output. The technical approach employed in developing SCAT and the rationale used in making the software and format decisions are reviewed, and a brief discussion of the important factors involved in the use of such a system for the preparation of computer authored programmed instructions is followed by recommendations. A glossary is provided, as well as five appendices containing listings of programs used by SCAT, typographer's substitution tables, a SCAT user's guide, code sets, and a sample of programmed instruction demonstrating the use of typography. (CHC)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

U S DEPARTMENT OF HEALTH, EDUCATION & WELFARE
NATIONAL INSTITUTE OF EDUCATION

THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL NATIONAL INSTITUTE OF EDUCATION POSITION OR POLICY

ED196399

TAEG Report No. 88

SYSTEM FOR COMPUTER AUTOMATED TYPESETTING (SCAT) OF COMPUTER AUTHORED TEXTS

F. Laurence Keeler

Training Analysis and Evaluation Group

July 1980

Sponsored by

Chief of Naval Education and Training

and the

David W. Taylor Naval Ship Research and Development Center,
Naval Technical Information Presentation Program

GOVERNMENT RIGHTS IN DATA STATEMENT

Reproduction of this publication in whole or in part is permitted for any purpose of the United States Government.

IR008840

Alfred F. Smode

ALFRED F. SMODE, Ph.D., Director,
Training Analysis and Evaluation Group

Worth Scanland

WORTH SCANLAND, Ph.D.
Assistant Chief of Staff for
Research and Instructional Systems
Development
Chief of Naval Education and Training

ACKNOWLEDGMENTS

The author is indebted to many people for support and guidance in completing this study. Special appreciation is extended to Mr. Charles R. Guitard of the Training Analysis and Evaluation Group (TAEG) for his generous assistance in the use of his subroutines to access the computer authored text created by the AUTHOR system. His general instruction and tutelage in basic programming has also been an immeasurable aid in this endeavor. The assistance of Dr. Richard Braby, Team Leader, and numerous other members of TAEG is also gratefully acknowledged.

Recognition is also extended to a number of people outside of TAEG for their assistance and counsel. The advice and tutelage in composition and typesetting by Mr. Frank Repca and Ms. Jerry Van Assche of Typographics and Mr. Douglas Minz (formerly of Typographics) have been invaluable and without which this project would not have been possible.

Special recognition and appreciation is extended to Intergraphics, Inc. for their permission to reprint the table of code sets contained in appendix D.

Finally, the author wishes to acknowledge the support of Dr. Alfred F. Smode, Director of TAEG.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER TAEG Report No. 88	2. GOVT ACCESSION NO	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) SYSTEM FOR COMPUTER AUTOMATED TYPESETTING (SCAT) OF COMPUTER AUTHORED TEXTS		5. TYPE OF REPORT & PERIOD COVERED 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) F. Laurence Keeler		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Training Analysis and Evaluation Group Orlando, FL 32813		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS		12. REPORT DATE July 1980 13. NUMBER OF PAGES 151
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution is unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Instructional Systems Development
Typesetting
Design of Instructional Materials
Computer Aided Authoring
Automated Authoring
Symbol Learning

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This study describes SCAT, an automated typesetting system for inserting special graphic symbols in programmed instructional materials. SCAT contains both hardware and software components and is especially designed for use with the computer authored texts created by the AUTHOR system described in TAEG Report No. 58 (Braby, Parrish, Guitard, and Aagard, 1978). It uses the digital data output of the AUTHOR system as its input and delivers camera-ready masters ready for the printer as its output. In addition to the

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. (continued)
inherent aesthetic quality resulting from its use, typesetting provides for the best possible legibility of the final product, whether it be hard copy or in the micromedia. However, the most important factor is economics. When a large number of nonstandard characters or symbols are required, such as in a programmed instruction for symbol learning, typesetting can eliminate the massive labor intensive step of hand stripping the symbols.

TABLE OF CONTENTS

Section		Page
I	INTRODUCTION.	3
	Background.	3
	Purpose	4
	Organization of this Report	4
II	OVERVIEW OF SCAT.	5
	The Composition Subsystem	5
	User Software.	5
	User Hardware.	5
	The Typesetting Subsystem	10
	Typographer's Software	10
	Typographer's Hardware	13
	Labor	13
III	SYSTEM DESIGN	15
	Functional Design	15
	Typesetting.	15
	Partitioning	15
	Telecommunications	16
	Function Allocation.	16
	Output Format Design.	17
IV	CONCLUSIONS AND RECOMMENDATIONS	19
GLOSSARY OF TECHNICAL TERMS USED IN THIS REPORT		21
REFERENCES.		27
APPENDIX A - LISTINGS OF PROGRAMS USED BY SCAT.		29
APPENDIX B - TYPOGRAPHER'S SUBSTITUTION TABLES.		89
APPENDIX C - SCAT USER'S GUIDE.		95
APPENDIX D - CODE SETS.		101
APPENDIX E - SAMPLE OF PROGRAMMED INSTRUCTION DEMONSTRATING THE USE OF TYPOGRAPHY.		107

LIST OF ILLUSTRATIONS

Figure		Page
1	Overview of SCAT.	6
2	Programmed Instruction Sample Page Listing.	8
3	Programmed Instruction Sample Intermediary Text	9
4	Programmed Instruction Sample Page in Typeset Form.	12

LIST OF TABLES

Table		Page
1	Composition Subsystem Components.	7
2	Typographer's Equipment	11

SECTION I

INTRODUCTION

This study describes SCAT, an automated system for typesetting text and inserting special graphic symbols in computer authored instructional materials. SCAT is designed to use the programmed instructional modules for symbol learning created by the AUTHOR system (Braby, Parrish, Guitard, and Aagard, 1978) as input and to output, ready for the printer, typeset camera-ready masters with symbols in place; thus eliminating the massive labor intensive step of hand stripping the symbols.

BACKGROUND

Publishing technical manuals and training materials for the Navy is expensive. In addition, current publishing practices frequently result in late delivery of technical data or delivery in a form unsuitable for the intended user. These publishing practices are also subject to problems in the revising and updating of technical documentation and in the storage, distribution, and maintenance of these materials.

The Naval Technical Information Presentation Program (NTIPP) of the David W. Taylor Naval Ship Research and Development Center is a large-scale effort to improve the Navy's efficiency in publishing technical information in support of Navy equipment. The goal of the program is to define the full family of hardware support documents required for equipment operation, maintenance, logistics support, and training, and to design a state-of-the-art system for authoring, composing, illustrating, mastering, replicating, distributing and updating these documents. Several Navy organizations are participating in this major effort. The Training Analysis and Evaluation Group (TAEG) was tasked by the Chief of Naval Education and Training (CNET) to represent the Naval Education and Training Command (NAVEDTRACOM) in this effort.

The present report is the fifth of a series of TAEG reports exploring ways of improving technical manuals and training materials. The first of these studies was a plan for CNET involvement with NTIPP for improving Navy training documents (Braby, 1977), while the second dealt with the use of microfiche as an instructional medium for technical training (Rizzo, 1977). The use of computer-based publishing techniques for meeting the publishing requirements of the CNET was the subject of the third study (Keeler, 1977). While various options in word processing, text editing, composing, and phototypesetting were studied, computer aids for the author were not included. The fourth study reports on the development of AUTHOR, a system for computer-aided authoring of programmed instruction for symbol recognition (Braby, Parrish, Guitard, and Aagard, 1978). While AUTHOR does provide powerful aids to the author, it does not include provision for typesetting or other automated preparation of camera-ready masters.

The study reported here has developed SCAT, a computer based system for typesetting the computer authored texts created by the AUTHOR system. This typesetting provides inherent aesthetic quality and high legibility, whether the medium is paper or micromedia. In addition, where large numbers of nonstandard symbols are required (as in a programmed instruction for symbol

learning), the SCAT eliminates the massive labor intensive task of pasting the symbols in by hand. Thus, when SCAT is used in conjunction with AUTHOR, the publishing of programmed instructions for symbol learning consumes hours rather than days of labor with concomitant dollar savings.

PURPOSE

This report describes SCAT, an automated system for typesetting special graphic symbols as well as the text of the programmed instructional materials created by AUTHOR, a computer aided authoring system. In addition, it outlines the design architecture of the system so those who have access to similar but not identical equipment will be able to modify the system for use in their environment.

ORGANIZATION OF THIS REPORT

Besides this introduction, the report contains three additional sections, a glossary, and five appendices. Section II contains an overview of the SCAT system, including component requirements, as well as a flow chart and examples of system input data, composed text stream, and typeset output. The technical approach employed in developing SCAT as well as the rationale used in making the software and format decisions is the subject of section III, which will be an aid to the reader should the system require modifying to accommodate his equipment limitations. In section IV, recommendations and conclusions are made following a discussion of the important factors impacting on the automated preparation of computer authored programmed instructions through the typesetting process. A glossary has been included as an aid to communications between the SCAT system user and the typographer.[1]

Listings of the software used by the COMPOSITION subsystem, which will be useful in the event of system alterations, are contained in appendix A. The substitution tables used by the typographer to make the transmitted code compatible with his part of the sytem are presented in appendix B. Appendix C is a user's guide for the operation and use of the SCAT system. A table of five standard data communications codes is reproduced in appendix D along with the binary, octal, hexadecimal, and decimal equivalents as an aid to the user in conversing with the typographer who may be using an alternative code to translate characters to decimal equivalents rather than hexadecimal. Finally, appendix E is a complete programmed instruction for teaching present weather symbols and demonstrates the quality of typography available.

[1] A discussion of typesetting in general is beyond the scope of this report. The reader not familiar with the current state of the art in typesetting will find Fundamentals of Modern Composition, Seybold, 1977, an excellent source.

Section II

OVERVIEW OF SCAT

This section provides an overview of SCAT. It is an automated system, consisting of both hardware and software, for typesetting the text and special graphic symbols contained in the programmed instructions created by the AUTHOR system. Because most users will not be able to afford a typesetter, the system is partitioned into two subsystems, the user's subsystem, COMPOSITION, and the typographer's subsystem, TYPESETTING. These two subsystems form the necessary links for transforming the digital form of the programmed instruction modules created by the AUTHOR system into camera-ready masters with symbols in place. Figure 1 shows the functional flow of the data as it is transformed from the programmed instruction created by the AUTHOR system into the typeset camera-ready masters ready for the printer.

THE COMPOSITION SUBSYSTEM

The COMPOSITION subsystem consists of the hardware and software necessary to transform the programmed instructional text into the composed text stream required as input by the typographer's subsystem. Table 1 lists the components of the COMPOSITION subsystem and figures 2 and 3 provide samples of the programmed text input and composed text stream output, respectively.

USER SOFTWARE. The Programmed Instruction disk file is the computer authored text created and stored by the AUTHOR system. While it is the output of the AUTHOR system, it is the input to the SCAT system. The Symbol Table is a data file containing a matrix of the symbol mnemonic codes employed by the AUTHOR system and the corresponding character command strings (consisting of from one to five characters) required by the typesetter to access the corresponding symbol. The Composed Text Stream is a file containing the text imbedded with the symbol character command strings required for the special symbols and the other required typesetting commands.

The executive program is an interactive program provided to initiate the functions to be performed by the user. While it is not a necessary part of the system, it decreases to a minimum the level of skills required to operate the system.

The three functional modules each contain the necessary software to perform the designated function. COMPOSE performs the actual text manipulation, transforming the Programmed Instruction file, in record format, into text stream format. Simultaneously it makes the required substitutions of character command strings for the symbol mnemonics, as dictated by the Symbol Table, and inserts the necessary typesetting commands. The second module, LIST, creates a lineprinter listing of the resultant Composed Text Stream file, which may be used for proofing the code before it is transmitted or for editing the transmitted text by the typographer in the event of a transmittal error. The final user module, TRANSMIT, sends the Composed Text Stream to the typographer over the commercial telephone lines.

USER HARDWARE. The hardware associated with the COMPOSITION subsystem is of the Wang 2200 series. The central processing unit (CPU), required for all

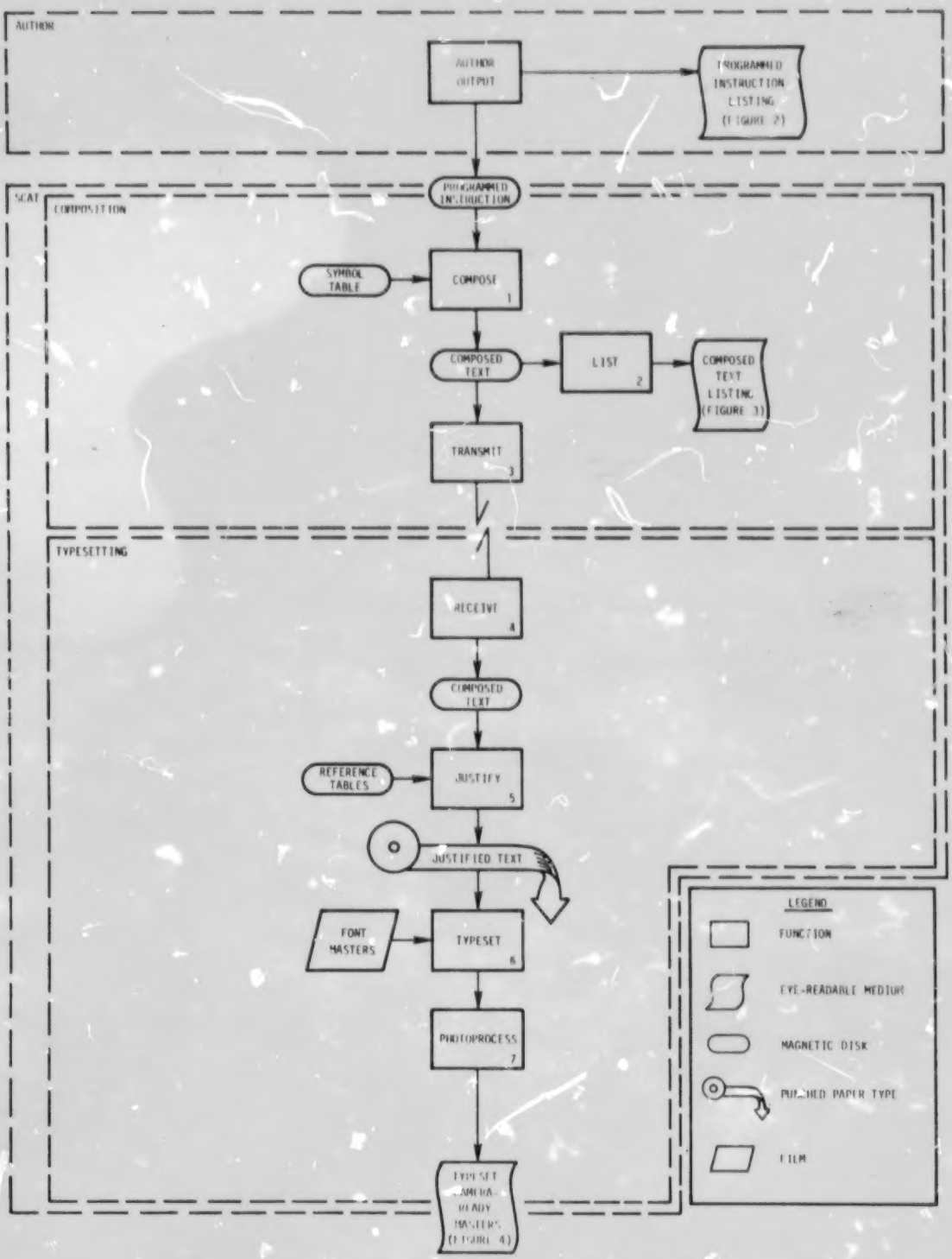


Figure 1. Overview of SCAT

TABLE 1. COMPOSITION SUBSYSTEM COMPONENTS

SOFTWARE	
Data Files	
Programmed Instruction	
Symbol Table	
Composed Text Stream	
Executive Program	
START	
Functional Modules	
COMPOSE	
LIST	
TRANSMIT	
HARDWARE	
Central Processing Unit:	Wang 2200 VP and Wang 2200 MVP (minimum 32K bytes of memory required)
Terminal:	Wang 2226 Keyboard/CRT
Printer:	Wang 2221 and Wang 2260 used interchangeably (Optional, required only for LIST function)
Mass Storage:	
Floppy Disk Drives	Wang 2270
Platter Disk Drives	Wang 2260 (Two disk drives required; the configuration used had 5 floppy and 2 platter disk drives available)
Telecommunications Interface:	
Controller	Wang 2227 Telecommunications Controller
Modem	Anderson Jacobson AD 342 Acoustic Data Access Coupler (Bell 103-A compatible)

```
'AUTHOR: An Automatic Authoring System -  Version:  7.0

OPTIONAL_CRITERION_TEST_____________________________  0032
Directions          1. Check your answers now.
                    2. Put an X through your wrong
                    answers.
Numbers___________ __________Answers_____________
 1. 7                %31. / %78.
 2. 4                %45.
 3. 6                %60.
 4. 9                %96.
 5. 5                %50.
 6. 7                %70.
 7. 8                %80.


                                        'Go to 0033
```

Figure 2. Programmed Instruction Sample Page Listing

```
§et
§el07
§bt
§tr
6§tc
%60.§tl
§et
§el07
§bt
§tr
9§tc
%96.§tl
§et
§el07
§bt
§tr
5§tc
%50.§tl
§et
§el07
§bt
§tr
7§tc
%70.§tl
§et
§el07
§bt
§tr
8§tc
%80.§tl
§et
§el07
§el16
§el07
§el16
§el07
§el16
§dt6g,10p,2g,21p,6p
§bt
§tl
§el15
§el99
§el16
Go to §g4
33§tr
§et
§el16
§qc
§el12
← ←§te
← ←§el12.
32§ql
§et
§el99
§dt6g,33p,6p
§bt
§g3
OPTIONAL►CRITERION►TEST§tl
§et
§el36
§bt
§li
§tj
§et
§el45
§dt6g,10p,2g,21p,6p
§el07
§bt
§g2
Directions§tl
§g1
1. Check your answers now.§tl
§et
§el07
§bt
§tl
2. Put an X through your wrong§tl
§et
§bt
§tl
answers.§tl
§et
§dt6g,3p,10p,20p,6p
§bt
§nt
§g2
Numbers§tc
Answers§tl
§et
§el07
§bt
§g1
§tr
3§tc
%31. / %38.§tl
§et
§el07
§bt
§tr
4§tc
%45.§tl
```

Figure 3. Programmed Instruction Sample Intermediary Text

three functions, must be a Wang 2200 VP or a Wang 2200 MVP because the concatenation instruction is used. The mass storage medium is used to store and access the three disk files as well as the functional programs and is also required for all three functions. The telecommunications controller and modem are required for the TRANSMIT function to get the Composed Text Stream to the typographer's subsystem, TYPESETTING. The lineprinter is needed for the LIST function in order to obtain a listing of the Composed Text Stream. Although this listing and, hence, the function and the lineprinter are not required, it is nice to have, and most computer installations will have at least one printer.

THE TYPESETTING SUBSYSTEM

The typographer's portion of the system, TYPESETTING, contains the hardware and software to transform the Composed Text Stream received from the COMPOSITION subsystem into the final typeset camera-ready masters required by the printer. Table 2 lists the components of the TYPESETTING subsystem, and figure 4 depicts a sample of the typeset output.[2]

TYPOGRAPHER'S SOFTWARE. The four functional modules each contain the necessary software for performing the indicated function using the associated hardware. RECEIVE receives the Composed Text Stream over commercial telephone lines and stores it in a disk file for later processing. JUSTIFY creates the Justified Text file on the punched paper tape which drives the typesetter using the Composed Text Stream as input and the Reference Tables for substituting typesetter peculiar commands and determining individual character widths. TYPESET performs the phototypesetting using the punched paper tape as input and exposing photosensitive paper (or film) with the specified characters at the prescribed locations as output. The exposed paper (film) is developed in the PHOTOPROCESS function, which passes it through chemical baths and washes and a dryer, thus making the camera-ready masters.

The Composed Text file, Reference Tables, and Justified Text file are digital data files. The Composed Text file holds the transmitted text and is identical to the Composed Text file in the COMPOSITION subsystem. The Reference Tables are of two types. The primary tables are the Width Tables which contain the individual character width information for each character in each font of type. The other set of Reference Tables are the "Find and ALTer" (FALT) Tables which are similar to the Symbol Table in the COMPOSITION subsystem. They are used to substitute the particular typesetting command strings or characters required to drive the actual typesetter used in place of the more generic commands generated by the COMPOSITION subsystem. The Justified Text file is created on a punched paper tape, because it drives the typesetter whose only input device is a punched paper tape reader.[3]

[2]These are the components of the subsystem used by the typographer to process the example shown. Because of different equipments or procedures, other typographers may accomplish the same results using a different complement of components.

[3]It should be noted that while a punched paper tape reader was the only input device for this typographer's typesetters, many typesetters have alternate inputs; e.g., magnetic tape, floppy disk, or direct input from the front end system computer.

TABLE 2. TYPOGRAPHER'S EQUIPMENT

SOFTWARE	
Functional Modules	
RECEIVE	
JUSTIFY	
TYPESET	
PHOTOPROCESS	
Data Files	
Composed Text File	
Reference Tables	
Justified Text File (driver tape)	
HARDWARE	
Telecommunications Interface	
Modem	Bell 103-A
Interface	Intergraphics Intercomm 100
Front End Composition System	Penta--Pentaware
Central Processing Unit	Data General - Nova II with 32K bytes of memory
Mass Storage	Diablo 10 Mega byte disk
Terminal	Bee Hive, Int.
Papertape Output	Data General Paper Tape Punch
Typesetter	Merganthaler VIP - Measurematic used interchangeably with Merganthaler VIP-HS
Film Processor	LogEtronics RAP 20

OPTIONAL CRITERION TEST

Directions

1. Check your answers now.
2. Put an X through your wrong answers.

Numbers	Answers
3	
4	
6	
9	
5	
7	
8	

Go to 33

Figure 4. Programmed Instruction Sample Page in Typeset Form

TYPOGRAPHER'S HARDWARE. The hardware associated with the TYPESETTING subsystem is divided into six major blocks. The telecommunications interface has a modem and a special purpose CPU which allows the typographer to receive and store the text to be typeset.[4] The Mass Storage devices consist of magnetic platter disk drives for storing and holding the Composed Text file and the Reference Tables, paper tape punches for punching the Justified Text file, and paper tape readers for inputting this data to the typesetters. The front end system is the part of the subsystem which massages the data into the form required to drive the typesetter. It consists essentially of a small minicomputer system with specialized software for typesetting, the usual peripherals--a terminal, disk storage, a printer, and a paper tape punch. The front end system usually composes and justifies the text, but because the input has already been composed by the COMPOSITION subsystem, only the justify function is performed. (Thus, the terminal and printer are not required for other than system control.) The typesetters receive the Justified Text through the paper tape readers and expose the typesetting medium (paper or film) to light in the required patterns as prescribed by the typesetting commands and mediated by the font masters. The font masters might be considered data files rather than hardware because it is the information as to where the typeset medium is to be exposed to light and where it is to remain unexposed that is important. Many typesetters maintain this information in digital form on magnetic disks. However, the typesetters used in this study use font masters contained on strips of film which are inserted into the typesetter. Therefore, the font masters have been included as hardware. Finally, the Photoprocessor passes the exposed typesetting medium through the necessary chemical baths and washes to develop the medium, then dries it, and finally outputs it as the typeset camera-ready masters.

LABOR

All of the operations performed by SCAT are highly automated and require a minimum of manual intervention. In the COMPOSITION subsystem, each function--COMPOSE, LIST, and TRANSMIT--requires only the depression of a few keys and switches for initiation, after which the function is accomplished unattended.[5] The total labor requirement of the COMPOSITION subsystem is only a few minutes per programmed instruction.

The functions performed by the TYPESETTING subsystem are also highly automated, except for the preparation of the required special font masters. Given original symbols in final art form, the typographer's charge is a flat $100 per font master. Since each font master may contain up to 96 graphic symbols, this averages to about $1.04 per unique symbol. Labor was negligible in the remaining TYPESET functions and the typographer charged only for the typesetter usage time. For the sample typeset symbol learning package contained in appendix E, this averaged $3.34 per page.

[4]This special purpose CPU and its associated software permit the typographer to receive and translate text to be typeset from a number of diverse sources, including word processors transmitting in their own codes as well as other CPUs transmitting in standard ASCII or EBCDIC codes.

[5]See the User's Guide in appendix C.

In another, more extensive test of the SCAT system an entire sample programmed instruction was typeset at a total cost of less than one man-hour of labor to perform the COMPOSITION functions plus $485 in charges by the typographer which included the cost of making the font master. The only additional equipment required, beyond that required by the AUTHOR system, was the telecommunications controller and modem.[6] Thus the 113 page programmed instruction was typeset at a cost of less than $4.50 per page. By contrast, using the traditional system, typing alone would probably have cost more and the hand insertion of over 1,200 symbols would have involved an additional 50 man-hours[7] of skilled labor.

[6] If this additional equipment is not already a part of the user's configuration, it can be rented for about $50 a month or purchased for $1,050.

[7] Based upon 2 to 3 minutes per symbol.

SECTION III

SYSTEM DESIGN

This section reviews the system design considerations and the rationale used in the development of SCAT. It is in two parts: the Functional Design, which addresses the functions required for producing the camera-ready masters with symbols in place, and the Output Format Design, which addresses the compositional form of this output.

FUNCTIONAL DESIGN

TYPESETTING. Ordinarily, when any graphics, such as photographs, line-drawings, or nonalphanumeric characters, are required to be printed, they are stripped (pasted) in by hand. This is costly and labor intensive but often unavoidable because typewriters and typesetters are not usually able to create other than standard alphanumeric characters. However, if a sufficiently large number of special symbols or characters are required, it becomes practical to consider having a special font master made for creating these symbols with a typesetter. (This would correspond to having a special "ball" made for a Selectric Typewriter.) Although this would not be practical on a typewriter because of the time and labor required to switch from one set of type to another, it is practical for a typesetter because it can switch between font masters automatically. The cost of creating font masters averages about $1 per symbol (or character). The cost of hand stripping a symbol is about $0.25 (assuming $6/hr. labor and 2-3 minutes/symbol). Since a symbol need be mastered only once if it is to be typeset, and the cost of stripping in the symbol every time it appears is eliminated, the mastering and typesetting method becomes economical when a symbol is to be used more than four times and other costs are not increased. For the symbol learning packages created by the AUTHOR system, each symbol is used an average of 25 times, giving some degree of confidence to the choice of typesetting this material.

PARTITIONING. The typesetting process requires inputting the text to be typeset, composing it, justifying it, and adding the typesetting commands necessary to drive the phototypesetter and photoprocessing the output film or paper. The phototypesetter is an expensive piece of equipment, requiring a trained operator, a photoprocessor, and special photosensitive materials. Its procurement is not recommended except for the high volume user. However, for the developer of training materials whose volume does not justify the acquisition of a typesetter, contracting typesetting requirements to a typographer is a viable alternative to traditional document preparation methods.

A typographer having a modern typesetter will usually have a minicomputer system for preparing the text to be typeset. This minicomputer system, called a front end system, is used for keying in the text, composing it, justifying it, and inserting the typesetting commands required for driving the phototypesetter. The major cost for typesetting is usually the labor required for keyboarding the text. Since the text created by the AUTHOR system is created in a computer compatible form, it is possible to dispense with the typographer's keyboarding step. This assumes that the user's computer and the typographer's computer share a compatible medium of data exchange.

TELECOMMUNICATIONS. The only compatible medium of data exchange available to the typographer chosen and the user was a 300-baud telecommunications link. The introduction of the telecommunications link added the requirement for a transmitting function to the user's tasks and a receiving function to the typographer's tasks.

FUNCTION ALLOCATION. In deciding which functions could best be performed by the user and which functions could best be left for the typographer's system to perform, it was necessary to consider the data format. The programmed instruction as created by the AUTHOR system is in record format; i.e., each page of instruction occupied a physical record block consisting of precisely 16 logical records of exactly 64 characters each, regardless of whether a page contained one word or 200 words. While this format is convenient for manipulating data files (as is done in creating the programmed instruction), the typographer's system is designed to operate on material which is in text stream format; i.e., terminating a line with an end-of-line command rather than filling it out with blanks and skipping blank lines rather than filling them with 64 blanks. (With proportionally spaced type, the number of characters which will fit on a line is not known until after the text has been justified.) Since the record format is a less efficient form for data transmission, where a large portion of the data may be blanks, the required conditioning of the programmed instruction to convert it from data record format to text stream format was allocated to the user's system.

In addition, the mnemonic codes used by the AUTHOR system to represent the special symbols were required to be converted to the typesetting command strings recognizable to the typesetter. The typographer could have performed this task because of the special telecommunications interface he had, but to more generalize the process to other typographers and because it could be readily accomplished using a symbol table, this conversion was allocated to the user.

Composing is still another task to be performed. Composing entails assigning where everything is to appear on the page and in what size and typeface it will be set. Although the AUTHOR system's output was "composed" to the extent of determining where everything was to appear, it did not indicate in what size or typeface the text was to be set because it was designed for lineprinter or typewriter output. This information is usually keyed in by the operator as the text is being keyboarded into the front end system. However, since the keyboarding step was being eliminated, other provisions were required. The conversion from record format to text stream format and the translation of the symbol mnemonic codes to typesetting command strings were already being done by the user's system. Also, the insertion of the generic typesetting commands to change from font to font and point size to point size could be more easily implemented as a part of the user's system than the typographer's production oriented system. Therefore, the reformatting of the textual data, the translation of the symbol codes, and the adding of the generic typesetting codes for composition were combined and performed simultaneously in a user function called COMPOSE.

A listing of the data output by the COMPOSE function was required for debugging purposes during system development. This listing function has been maintained as a proofing tool and for use in editing the transmitted text in

the event of an error in transmission. It is called LIST and has been included as a separate function because in the user's multiuser environment, a lineprinter may not be available at all times when a terminal is available for performing other functions.

The remaining user function is that required to transmit the data to the typographer's system. Maintaining this function, TRANSMIT, as a separate function minimizes the hardware required to be available at the same time the typographer is ready to receive the data transmittal.

The justifying function which fills the typeset line or measure with text requires width information for each character of each font in each point size used and relatively sophisticated software to implement it. This function was therefore allocated to the typographer's system where the necessary data and software already existed. This function, called JUSTIFY, also included the insertion of typesetting commands peculiar to the typographer's typesetters.

The actual typesetting and photoprocessing functions were allocated to the typographer's systems and called TYPESET and PHOTOPROCESS, respectively.

To summarize, the functions of COMPOSE, LIST, and TRANSMIT made up the subsystem called COMPOSITION and were allocated to the user's system. The functions of RECEIVE, JUSTIFY, TYPESET, and PHOTOPROCESS made up the subsystem called TYPESETTING, and were allocated to the typographer's system.

OUTPUT FORMAT DESIGN

The output format, with few exceptions, maintains the line breaks and page breaks made by the AUTHOR system, which preserves the lineprinter output of the AUTHOR system as a close approximation of the final typeset output. Although this has resulted in the preservation of some widows and orphans[8] which might otherwise have been eliminated through good composition practices, it was necessitated by the numerous page references contained in the programmed text. (To change these would have required repeating many of the functions performed by the AUTHOR system.) Preserving the line breaks generated by the AUTHOR system also avoids excessive compression of the text toward the top of a page which would otherwise occur in switching to proportionally spaced type.[9] Exceptions to this rule have occurred where emboldening of text has caused a line to exceed its measure (i.e., the width assigned to it) but was otherwise adhered to with the result that the text, set ragged-right,[10] is sometimes exceptionally ragged.

[8] When a paragraph has been broken between two pages so that one page contains only one line, the isolated line is referred to as a "widow." When a paragraph ends with only one word on the last line, the isolated word is referred to as an "orphan." Both widows and orphans are anathema to compositors and typographers.

[9] Approximately 25 percent more characters may be set on a line of proportionally-spaced type as can be set in monospaced typewriter or lineprinter type in the same type size.

[10] Typeset material which is set with an uneven (ragged) right-hand margin is referred to as being set "ragged-right."

Outside of the constraints on line breaks and pagination mentioned above, the only limitations to composition were those imposed by automation and good composition practice. Thus, page heads were set in Helvetica Bold, side heads were set in Century Schoolbook Bold, and the body text was set in Century Schoolbook. (Because the final user medium would sometimes be in the micromedia, the use of nonconventional or highly stylized typefaces was avoided.) To facilitate plate making, cut marks were set at the top and bottom of each page. To facilitate frame location in the programmed instruction, page numbers were located differently for microfiche and hard copy. When the medium was microfiche, the frame number was set in the middle of both the bottom and top of each frame. Thus the current frame number would always be visible even if the frame were poorly aligned. When the medium was hard copy, the page numbers were set at the upper right corner of odd numbered pages and upper left corner of even numbered pages.

SECTION IV

CONCLUSIONS AND RECOMMENDATIONS

The present study has demonstrated significant labor savings and concomitant dollar savings through the use of the System for Computer Automated Typesetting (SCAT) when nonalphanumeric characters or symbols are required in the camera-ready masters of a computer authored text. In addition, the sample of programmed instruction produced by SCAT (see appendix E) indicates the high aesthetic quality inherent in typeset material (independent of the symbols). Finally, the study has shown that SCAT is an easily implementable system for use in conjunction with the computer authored symbol learning packages created by the AUTHOR system. Accordingly, SCAT:

- is cost effective in preparing the camera-ready masters whenever nonalphanumeric characters are required
- improves the aesthetic quality of computer authored texts because the output is typeset
- provides an economical and easily implemented means for typesetting computer authored texts by separating the composing and typesetting functions.

Although the use of SCAT or a similar automated typesetting system is clearly indicated, the final form of an automated system is not as firmly established. It is possible that further material savings could accrue through the use of a more sophisticated compositor system. Such a system would make more effective use of proportionally spaced type, thus producing additional savings. This might be best accomplished by incorporating the composition subsystem as a part of the AUTHOR system. Alternatively, employing distributed processing techniques in the AUTHOR system with the typographer's front end system merging and composing the programmed text from the AUTHOR system's data base should also be considered.

Based on these observations, the following is recommended:

- the SCAT or a similar system be used to automatically typeset instructional materials for symbol learning created by the AUTHOR system
- a SCAT-like system be incorporated into the AUTHOR system to accommodate instructional material for procedure learning to be developed by the next generation AUTHOR system.

GLOSSARY OF
TECHNICAL TERMS USED IN THIS REPORT

GLOSSARY

ASCII	American National Standard Code for Information Interchange--a common 7-bit digital computer code.
baud	A metric of the speed at which signals may be transmitted. 300 baud equals 300 bits per second in a train of binary bits, or about 30 alphanumeric characters per second.
boilerplate	A set of repetitive blocks of text which may be included routinely without the need to be re-created.
Camera-ready masters	Textual material to be printed in a form from which the printer can make a photoengraved printing plate.
CNET	Chief of Naval Education and Training
CPU	Central Processing Unit--the heart of a digital computer which performs or commands the performance of the programmed instructions.
CRT Display/ Keyboard	Cathode Ray Tube Display/Keyboard--a device resembling a small television monitor with a typewriter keyboard used to input information to the CPU. Also frequently referred to as a Video Display Terminal (VDT).
em	A variable metric, usually about the width of an uppercase "M" in the type face and point size being set.
em-space	A fixed space of one em in width. Sometimes called a mutt.
en	A variable metric usually about ½ the width of an em, but most frequently the width of the digits in the type face and point size being set.
en-space	A fixed space one en in width. Sometimes called a nut or figure space.
FALT Table	Find and ALTer table--name assigned to Penta's substitution tables.
font	A standard assortment of signs of a particular graphic design.

GLOSSARY (continued)

font master	A set of coded information used for producing the signs of a given font. In phototypesetters, one font master may be used to set characters over a range of sizes by varying the magnification used in the optical path of the typesetter.
graphic	Anything other than standard alphanumeric characters to be printed.
justified	Composed lines of text where the left and right margins are even. This is accomplished by adjusting the variable spacebands. See ragged.
leadering	The filling of a portion of the line measure with a repeated character, most often a string of periods. (Rhymes with metering.)
leading	A measure of the vertical distance between horizontal rows of text. (Rhymes with heading.)
measure	The set width of a line, usually specified in points.
modem	Modulator/Demodulator--an electronic device for converting digital data to an analog signal (modulating) which may be transmitted over commercial telephone lines and for converting the transmitted analog signal back to digital form (demodulating) for processing by a remote digital device.
monospaced type	Type which is characterized by having all characters take up the same horizontal space, regardless of the width actually required. See proportionally spaced type.
NTIPP	Naval Technical Information Presentation Program
orphan	An isolated single word appearing on the last line of a paragraph. Orphans are anathema to compositors and typographers.
photoprocessor	An electro-chemical-mechanical device for completing the process initiated by the phototypesetter in the preparation of camera-ready masters or micromedia masters. Also referred to simply as a processor.
phototypesetter	An electro-opto-mechanical device for creating the exposed film for camera-ready masters.
processor	See photoprocessor.

GLOSSARY (continued)

proportionally spaced type	Type in which each character has a width equal to that which is required to print it. See monospaced type.
quad center	A composing command which causes the typesetter to center the text in the line measure and fill out both ends with white space. This is most often used for centering headings.
quad left	A composing command which causes the typesetter to place the text at the left of the line and to fill out the line measure to the right with white space. Successive lines of this result in text set ragged-right.
quad right	A composing command which causes the typesetter to place the text at the right of the line measure and to fill out the line measure to the left with white space. Successive lines result in text set ragged-left.
ragged	Composed lines of text where the margins are not even. See quad left and quad right.
RAM	Random Access Memory--usually a solid state memory which is volatile; i.e., is lost when power is removed and must be re-loaded when power is restored.
reverse leading	The ability of a typesetter to move the typeset medium in the opposite direction. Often used in order to set a second column along side a first.
sans-serif	Type which does not have serifs. See serif.
serif	The short lines stemming from and at an angle to the upper and lower ends of the strokes of a character. Also type whose characters have serifs.
spaceband	A space of indeterminate width (though nominally about 1/3 of an em-space) which may be varied to help justify a line.
Stripping	The hand pasting of strips of typeset text into the final camera-ready form, and by extension, the hand pasting of any materials (e.g., nonstandard characters or logos) to complete the camera-ready master.
TAEG	Training Analysis and Evaluation Group
TTS	TeleTypeSetting Code--a 6-bit communications code especially designed for transmitting text using paper-tape.

GLOSSARY (continued)

type face	a particular design of type characters.
typesetter	See phototypesetter.
widow	An isolated first or last line of a paragraph appearing at the bottom or top of a page when a paragraph transcends two pages. Widows are anathema to compositors.

REFERENCES

Braby, Richard. Training Requirements for the Naval Technical Information Presentation Program: A Needs Assessment. TAEG Technical Memorandum 77-3, 1977. Training Analysis and Evaluation Group, Orlando, FL 32813 (ADA040260).

Braby, Richard, Parrish, William F., Guitard, Charles R., Aagard, James A. Computer-Aided Authoring of Programmed Instruction for Teaching Symbol Recognition. TAEG Report No. 58, 1978. Training Analysis and Evaluation Group, Orlando, FL 32813 (ADA060143).

Cornog, D. Y. and Rose, F. C. Legibility of Alphanumeric Characters and Other Symbols. II A Reference Handbook. Miscellaneous 262-2, 1967. National Bureau of Standards.

Guitard, Charles R. Computer-Aided Authoring System (AUTHOR) User's Guide. TAEG Report No. 59, 1978. Training Analysis and Evaluation Group, Orlando, FL 32813 (Vol. I: ADA059572; Vol. II: ADA061611).

Hartley, James H. Designing Instructional Text. New York: Nichols Publishing Co. 1979.

Hartley, James H. The Psychology of Written Communications: Selected Readings. New York: Nichols Publishing Co. 1980.

Interservice Procedures for Instructional Systems Development, Phase III, Develop. NAVEDTRA 106A. August 1, 1975. Chief of Naval Education and Training, Pensacola, FL 32508.

Keeler, F. Laurence. An Automated Publishing System for the Naval Education and Training Command. TAEG Report No. 50, 1977. Training Analysis and Evaluation Group, Orlando, FL 32813 (ADA047648).

Rizzo, William F. Evaluation of Microfiche as an Instructional Medium in a Technical Training Environment. TAEG Report No. 48, 1977. Training Analysis and Evaluation Group, Orlando, FL 32813 (ADA044519).

Seybold, John W. Glossary and Guide for Automated Typesetting. Media, PA: Seybold Publications, Inc. 1975.

Seybold, John W. Fundamentals of Modern Composition. Media, PA: Seybold Publications, Inc. 1977.

APPENDIX A

LISTINGS OF PROGRAMS USED BY SCAT

START

This program is the executive program of the COMPOSE subsystem. It calls and loads the other user programs used by the system.

START

```
 0000 $PSTAT = "CONSOLE"
 0008 COM S1$(128)4,S3$(128)13
 0009 DIM M9%3,M8%3,M7%3,S9%8,S8%8,S7%8
    : REM      : mass Storage addresses
 0010 %    ABSTRACT
-0011 % This program is designed to act as an executive for the remaining TYPESET progress
      . It displays a menu of functions available to the TYPESET system and loads the prog
      rams necessary for performing the selected function.
-0012 % Touch RETURN to display menu ...
-0013 %The following functions are available to the TYPESET system:
-0014 % Select the desired function by depressing the FUNCTION KEY corresponding to the fu
      nction to be performed.
-0020 %   FUNCTION KEY                    FUNCTION DESCRIPTION
-0021 %         1          MODIFY symbol Translation Table Mnemonic Codes.
-0022 %         2          COMPOSE the Computer Authored Text.
-0023 %         3          LIST the Composed Text.
-0024 %         4          TRANSMIT the Composed Text.
-0025 %        25          ALTER symbol Translation Table Typesetting Codes.
 0030 PRINT HEX(03)
    : PRINTUSING 11
    : PRINT
    : PRINTUSING 12
    : KEYIN  S9$
-0040 PRINT HEX(03)
    : PRINTUSING 13
    : PRINT
    : PRINTUSING 20
    : PRINT
    : PRINT
    : PRINTUSING 21
    : PRINT
    : PRINTUSING 22
    : PRINT
    : PRINTUSING 23
    : PRINT
    : PRINTUSING 24
    : PRINT
    : PRINT
    : PRINTUSING 25
    : PRINT
    : PRINTUSING 14
    : KEYIN S9$
 0050 ON VAL(S9$)GOTO 51,52,53,54
    : ON VAL(S9$)-24 GOTO 900
    : ON VAL(S9$)-48 GOTO 51,52,53,54
    : PRINT HEX(07)
    : GOTO 40

-0051 GOSUB '1
    : GOTO 40

-0052 GOSUB '2
    : GOTO 40

-0053 GOSUB '3
    : GOTO 40

-0054 GOSUB '4
    : GOTO 40

 0059 GOSUB '9
    : GOTO 40

 0100 DEFFN'1
 0110 PRINT HEX(03)
    : PRINTUSING 111
    : PRINT
    : PRINTUSING 112
    : KEYIN S0$
    : PRINT
    : PRINTUSING 113,A0$;
    : INPUT A0$
    : GOSUB '100(0,A0$)
    : N0$="TRANSLAT"
    : PRINT
    : PRINTUSING 114,N0$;
    : INPUT N0$
```

```
    :                    DATA LOAD DC S1$().S3$()
-0111 % Load the disk containing the table to be modifed and depress RETURN.
-0112 %      NOTE: The standard translation table. TRANSLAT. is on the TYPESET
            system disk.
-0113 %Input address of the disk containing the table to be modified (###) ...
-0114 %Input name of the table to be modified (########) ...
 0119 PRINT HEX(03)
-0120 PRINT AT(15,0,640)
    : PRINT
    : PRINTUSING 128
    : PRINTUSING 129
    : U=0
    : INPUT "           ",U
    : IF U=0 THEN U=I+1
    : IF U > 0 AND U < 100 THEN 130
    : REM ERROR
    : PRINT HEX(07)
    : GOTO 120

-0128 %Input the Number of the Mnemonnic Code to be altered.
-0129 %       NOTE: Entering a blank will increment the count to the next code.
-0130 I = U
    : PRINT AT(14,0,640)
    : PRINT AT(16,10);I
    : PRINT ,S1$(I),S3$(I)
    : PRINTUSING 138
    : PRINTUSING 139
    : U$=" "
    : INPUT "                ",U$
    : IF LEN(U$) = 4 THEN S1$(I)=U$
    : PRINT AT(24,0)
    :    PRINT AT(13,10);I,S1$(I),S3$(I)
    : PRINT AT(0,0,320)
    : PRINTUSING 136
    : PRINT
    : PRINTUSING 137
    : GOTO 120

-0136 %Depress SPECIAL FUNCTION KEY 15 when table modification is complete.
-0137 %           NO.  MNEMONIC CODE   TYPESETTING CODE
-0138 %To change Mnemonic Code, INPUT a NEW 4-character Mnemonic Code.
-0139 %      NOTE: Any input containing other than 4-characters will be ignored.
 0199 DATA SAVE DC S1$(),S3$()
    : DATA SAVE DC CLOSE
    : RETURN

-0200 DEFFN'2
    : PRINT HEX(03)
    : PRINT "COMPOSE"
    : PRINTUSING 201
    : KEYIN S0$
    : PRINTUSING 202,A0$
    : INPUT A0$
    : GOSUB '100(1,A0$)
    : IF N=-1THEN 200
    : PRINTUSING 203,N0$
    : INPUT N0$
    : DATA LOAD DC OPEN T#1,N0$
    : DATA LOAD DC S1$(),S3$()
-0201 %Load disk containing translation table to be used and depress RETURN.
-0202 %Input address of disk containing table to be used (###) ...
-0203 %Input name of translation table to be used (########) ...
-0210 PRINTUSING 211
    : KEYIN S0$
    : PRINTUSING 212,A0$
    : INPUT A0$
    : GOSUB '100(1,A0$)
    : IF N=-1THEN 200
    : N0$="COMPOSED"
    : PRINTUSING 213,N0$
    : INPUT N0$
-0211 %Load disk on which Composed Text is to be stored and depress RETURN.
-0212 %Input address of disk on which Composed Text is to be stored (###) ...
-0213 %Input name Composed Text is to be stored under (########) ...
 0220 DATA LOAD DC OPEN T#2,N0$
    : ERROR GOTO 230
    : PRINTUSING 221,N0$
    : S0$=" "
    : INPUT S0$
    : IF S0$<>"Y"THEN 210
```

```
    : GOTO 240

-0230 PRINTUSING 231
    : N=100
    : INPUT N
    : N=5*N
    : DATA SAVE DC OPEN T #2,(N),N0$
-0231 %Input number of pages to be composed ...
-0240 LOAD T"TYPE.INI"

    : LOAD T"TYPE.TST",BEG 1000

    : GOTO 40

-0300 DEFFN'3
    : PRINT HEX(03)
    : PRINT "LIST"
    : PRINTUSING 301
    : KEYIN S0$
    : PRINTUSING 302,A0$
    : INPUT A0$
    : GOSUB '100(2,A0$)
    : IF N=-1THEN 300
    : PRINTUSING 303,N0$
    : INPUT N0$
    : DATA LOAD DC OPEN T#2,N0$
    : ERROR GOTO 300
    : LOAD T"LIST.IT",BEG1000

    : GOTO 40

-0301 %Load disk on which Composed Text is stored and depress RETURN.
-0302 %Input address of disk from which Composed Text is to be listed (###)...
-0303 %Input name Composed Text to be listed is stored under (########) ...
-0400 DEFFN'4
    : PRINT HEX(03)
    : PRINT "TRANSMIT"
    : PRINTUSING 301
    : KEYIN S0$
    : PRINTUSING 402,A0$
    : INPUT A0$
    : GOSUB '100(2,A0$)
    : IF N=-1THEN 400
    : PRINTUSING 403,N0$
    : INPUT N0$
    : DATA LOAD DC OPEN T#2,N0$
    : ERROR GOTO 400
    : LOAD T"SEND.IT",BEG1000

    : GOTO 40

-0402 %Input address of disk from which Composed Text is to be transmitted (###)
-0403 %Input name Composed Text to be transmitted is stored under (########) ...
 0500 DEFFN'100(N,A0$)
    : IF POS("3BD"=STR(A0$,1,1))*POS("123567"=STR(A0$,2,1))*POS("012345"=STR(A0$,3,1))<>0
      THEN SELECT #N<A0$>
    : ELSE N=-1
    : RETURN

-0900 DEFFN'9
 0910 PRINT HEX(03)
    : PRINTUSING 111
    : PRINT
    : PRINTUSING 112
    : KEYIN S9$
    : PRINT
    : PRINTUSING 113,A0$;
    : INPUT A0$
    : GOSUB '100(0,A0$)
    : A0$="TRANSLAT"
    : PRINT
    : PRINTUSING 114,A0$;
    : INPUT A0$
    : DATA LOAD DC OPEN TA0$
                        DATA LOAD DC S1$().S3$()
-0920 PRINT AT(15,0,640)
    : PRINT
    : PRINTUSING 928
    : PRINTUSING 129
    : U=0
```

```
    : INPUT "          ",U
    : IF U=0 THEN U=I+1
    : IF U > 0 AND U < 100 THEN 130
    : GOTO 990

-0928 %Input the Number of the Typesetting Code to be altered.
 0930 I = U
    : PRINT AT(14,0,640)
    : PRINT AT(16,10);I
    : PRINT ,S1$(I),S3$(I)
    : PRINTUSING 938
    : PRINTUSING 939
    : U$=" "
    : INPUT "                ",U$
    : IF LEN(U$) >13 THEN GOTO 990
    : IF U$<>" "THEN S3$(I)=U$
    : PRINT AT(24,0)
    :    PRINT AT(13,10);I,S1$(I),S3$(I)
    : PRINT AT(0,0,320)
    : PRINTUSING 136
    : PRINT
    : PRINTUSING 137
    : GOTO 120

-0938 %To change Typesetting Code, INPUT the NEW Typesetting Code.
-0939 %   NOTE: A BLANK will result in no change. More than 13-characters is illegal.
-0990 REM ERROR
    : PRINT HEX(07)
    : GOTO 920

-1500 DEFFN'15
    : PRINT HEX(03)
    : PRINTUSING 1501
    : INPUT S0$
    : IF S0$="N" THEN 40
    : IF S0$<>"S"THEN 1500
    : PRINTUSING 1502,A0$
    : INPUT A0$
    : GOSUB '100(0,A0$)
    : IF N=-1THEN 1500
    : PRINTUSING 1503,N0$
    : INPUT N0$
-1501 %Input "S" if new table is to be saved: otherwise input "N":
-1502 %Input disk address of disk on which new table is to be stored (###) ...
-1503 %Input name of new table (########) ...
 1510 DATA LOAD DC OPEN TN0$
    : ERROR GOTO 1520
    : PRINTUSING 1511,N0$
    : S0$=" "
    : INPUT S0$
    : IF S0$<>"Y"THEN 1500
    : DATA SAVE DC S1$(),S3$()
    : DATA SAVE DC END
    : GOTO 40

-1511 %"##########" is already catalogued. do you wish to OVERWRITE it (Y.N)
-1520 DATA SAVE DC OPEN T(10) N0$
    : DATA SAVE DC S1$(),S3$()
    : DATA SAVE DC END
    : GOTO 40
```

START

```
 0000 $PSTAT = "CONSOLE"
 0008 COM S1$(128)4,S3$(128)13
 0009 DIM M9%3,M8%3,M7%3,S9%8,S8%8,S7%8
    : REM      : mass Storage addresses
 0010 %    ABSTRACT
-0011 % This progrss is designed to act as an executive for the remaining TYPESE
      T progrsss. It displays a menu of functions available to the TYPESET syste
      m and loads the progrsss necessary for performing the selected function.
-0012 % Touch RETURN to display menu ...
-0013 %The following functions are available to the TYPESET system:
-0014 % Select the desired function by depressing the FUNCTION KEY corresponding
       to the function to be performed.
-0020 %    FUNCTION KEY                    FUNCTION DESCRIPTION
-0021 %          1              MODIFY symbol Translation Table Mnemonic Codes.
-0022 %          2              COMPOSE the Computer Authored Text.
-0023 %          3              LIST the Composed Text.
-0024 %          4              TRANSMIT the Composed Text.
-0025 %         25              ALTER symbol Translation Table Typesetting Codes.
 0030 PRINT HEX(03)
    : PRINTUSING 11
    : PRINT
    : PRINTUSING 12
    : KEYIN  S9$
-0040 PRINT HEX(03)
    : PRINTUSING 13
    : PRINT
    : PRINTUSING 20
    : PRINT
    : PRINT
    : PRINTUSING 21
    : PRINT
    : PRINTUSING 22
    : PRINT
    : PRINTUSING 23
    : PRINT
    : PRINTUSING 24
    : PRINT
    : PRINT
    : PRINTUSING 25
    : PRINT
    : PRINTUSING 14
    : KEYIN S9$
 0050 ON VAL(S9$)GOTO 51,52,53,54
    : ON VAL(S9$)-24 GOTO 900
    : ON VAL(S9$)-48 GOTO 51,52,53,54
    : PRINT HEX(07)
    : GOTO 40

-0051 GOSUB '1
    : GOTO 40

-0052 GOSUB '2
    : GOTO 40

-0053 GOSUB '3
    : GOTO 40

-0054 GOSUB '4
    : GOTO 40

 0059 GOSUB '9
    : GOTO 40

 0100 DEFFN'1
 0110 PRINT HEX(03)
    : PRINTUSING 111
    : PRINT
    : PRINTUSING 112
    : KEYIN S0$
    : PRINT
    : PRINTUSING 113,A0$;
    : INPUT A0$
    : GOSUB '100(0,A0$)
    : N0$="TRANSLAT"
    : PRINT
    : PRINTUSING 114,N0$;
    : INPUT N0$
    : DATA LOAD DC OPEN TN0$
```

```
    :                    DATA LOAD DC S1$().S3$()
-0111 % Load the disk containing the table to be modifed and depress RETURN.
-0112 %      NOTE: The standard translation table. TRANSLAT. is on the TYPESET
                   system disk.
-0113 %Input address of the disk containing the table to be modified (###) ...
-0114 %Input name of the table to be modified (########) ...
 0119 PRINT HEX(03)
-0120 PRINT AT(15,0,640)
    : PRINT
    : PRINTUSING 128
    : PRINTUSING 129
    : U=0
    : INPUT *            *,U
    : IF U=0 THEN U=I+1
    : IF U > 0 AND U < 100 THEN 130
    : REM ERROR
    : PRINT HEX(07)
    : GOTO 120

-0128 %Input the Number of the Mnemonnic Code to be altered.
-0129 %       NOTE: Entering a blank will increment the count to the next code.
-0130 I = U
    : PRINT AT(14,0,640)
    : PRINT AT(16,10);I
    : PRINT ,S1$(I),S3$(I)
    : PRINTUSING 138
    : PRINTUSING 139
    : U$=" "
    : INPUT *                *,U$
    : IF LEN(U$) = 4 THEN S1$(I)=U$
    : PRINT AT(24,0)
    :    PRINT AT(13,10);I,S1$(I),S3$(I)
    : PRINT AT(0,0,320)
    : PRINTUSING 136
    : PRINT
    : PRINTUSING 137
    : GOTO 120

-0136 %Depress SPECIAL FUNCTION KEY 15 when table modification is complete.
-0137 %              NO.  MNEMONIC CODE   TYPESETTING CODE
-0138 %To change Mnemonic Code, INPUT a NEW 4-character Mnemonic Code.
-0139 %      NOTE: Any input containing other than 4-characters will be ignored.

 0199 DATA SAVE DC S1$(),S3$()
    : DATA SAVE DC CLOSE
    : RETURN

-0200 DEFFN'2
    : PRINT HEX(03)
    : PRINT "COMPOSE"
    : PRINTUSING 201
    : KEYIN S0$
    : PRINTUSING 202,A0$
    : INPUT A0$
    : GOSUB '100(1,A0$)
    : IF N=-1THEN 200
    : PRINTUSING 203,N0$
    : INPUT N0$
    : DATA LOAD DC OPEN T#1,N0$
    : DATA LOAD DC S1$(),S3$()
-0201 %Load disk containing translation table to be used and depress RETURN.
-0202 %Input address of disk containing table to be used (###) ...
-0203 %Input name of translation table to be used (########) ...
-0210 PRINTUSING 211
    : KEYIN S0$
    : PRINTUSING 212,A0$
    : INPUT A0$
    : GOSUB '100(1,A0$)
    : IF N=-1THEN 200
    : N0$="COMPOSED"
    : PRINTUSING 213,N0$
    : INPUT N0$
-0211 %Load disk on which Composed Text is to be stored and depress RETURN.
-0212 %Input address of disk on which Composed Text is to be stored (###) ...
-0213 %Input name Composed Text is to be stored under (########) ...
 0220 DATA LOAD DC OPEN T#2,N0$
    : ERROR GOTO 230
    : PRINTUSING 221,N0$
    : S0$=" "
    : INPUT S0$
```

```
    : IF S0$<>"Y"THEN 210
    : GOTO 240

-0230 PRINTUSING 231
    : N=100
    : INPUT N
    : N=5*N
    : DATA SAVE DC OPEN T #2,(N),N0$
-0231 %Input number of pages to be composed ...
-0240 LOAD T"TYPE.INI"

    : LOAD T"TYPE.TST",BEG 1000

    : GOTO 40

-0300 DEFFN'3
    : PRINT HEX(03)
    : PRINT "LIST"
    : PRINTUSING 301
    : KEYIN S0$
    : PRINTUSING 302,A0$
    : INPUT A0$
    : GOSUB '100(2,A0$)
    : IF N=-1THEN 300
    : PRINTUSING 303,N0$
    : INPUT N0$
    : DATA LOAD DC OPEN T#2,N0$
    : ERROR GOTO 300
    : LOAD T"LIST.IT",BEG1000

    : GOTO 40

-0301 %Load disk on which Composed Text is stored and depress RETURN.
-0302 %Input address of disk from which Composed Text is to be listed (###)...
-0303 %Input name Composed Text to be listed is stored under (########) ...
-0400 DEFFN'4
    : PRINT HEX(03)
    : PRINT "TRANSMIT"
    : PRINTUSING 301
    : KEYIN S0$
    : PRINTUSING 402,A0$
    : INPUT A0$
    : GOSUB '100(2,A0$)
    : IF N=-1THEN 400
    : PRINTUSING 403,N0$
    : INPUT N0$
    : DATA LOAD DC OPEN T#2,N0$
    : ERROR GOTO 400
    : LOAD T"SEND.IT",BEG1000

    : GOTO 40

-0402 %Input address of disk from which Composed Text is to be transmitted (###)

-0403 %Input name Composed Text to be transmitted is stored under (########) ...

 0500 DEFFN'100(N,A0$)
    : IF POS("3BD"=STR(A0$,1,1))*POS("123567"=STR(A0$,2,1))*POS("012345"=STR(A0$
      ,3,1))<>0 THEN SELECT #N<A0$>
    : ELSE N=-1
    : RETURN

-0900 DEFFN'9
 0910 PRINT HEX(03)
    : PRINTUSING 111
    : PRINT
    : PRINTUSING 112
    : KEYIN S9$
    : PRINT
    : PRINTUSING 113,A0$;
    : INPUT A0$
    : GOSUB '100(0,A0$)
    : A0$="TRANSLAT"
    : PRINT
    : PRINTUSING 114,A0$;
    : INPUT A0$
    : DATA LOAD DC OPEN TA0$
    :                    DATA LOAD DC S1$().S3$()
-0920 PRINT AT(15,0,640)
    : PRINT
```

```
    : PRINTUSING 928
    : PRINTUSING 129
    : U=0
    : INPUT "            ",U
    : IF U=0 THEN U=I+1
    : IF U > 0 AND U < 100 THEN 130
    : GOTO 990

-0926 %Input the Number of the Typesetting Code to be altered.
 0930 I = U
    : PRINT AT(14,0,640)
    : PRINT AT(16,10);I
    : PRINT ,S1$(I),S3$(I)
    : PRINTUSING 938
    : PRINTUSING 939
    : U$=" "
    : INPUT "                ",U$
    : IF LEN(U$) >13 THEN GOTO 990
    : IF U$<>" "THEN S3$(I)=U$
    : PRINT AT(24,0)
    :    PRINT AT(13,10);I,S1$(I),S3$(I)
    : PRINT AT(0,0,320)
    : PRINTUSING 136
    : PRINT
    : PRINTUSING 137
    : GOTO 120

-0938 %To change Typesetting Code, INPUT the NEW Typesetting Code.
-0939 %   NOTE: A BLANK will result in no change. More than 13-characters is ill
      egal.
-0990 REM ERROR
    : PRINT HEX(07)
    : GOTO 920

-1500 DEFFN'15
    : PRINT HEX(03)
    : PRINTUSING 1501
    : INPUT S0$
    : IF S0$="N" THEN 40
    : IF S0$<>"S"THEN 1500
    : PRINTUSING 1502,A0$
    : INPUT A0$
    : GOSUB '100(0,A0$)
    : IF N=-1THEN 1500
    : PRINTUSING 1503,N0$
    : INPUT N0$
-1501 %Input "S" if new table is to be saved: otherwise input "N":
-1502 %Input disk address of disk on which new table is to be stored (###) ...
-1503 %Input name of new table (########) ...
 1510 DATA LOAD DC OPEN TN0$
    : ERROR GOTO 1520
    : PRINTUSING 1511,N0$
    : S0$=" "
    : INPUT S0$
    : IF S0$<>"Y"THEN 1500
    : DATA SAVE DC S1$(),S3$()
    : DATA SAVE DC END
    : GOTO 40

-1511 %"##########" is already catalogued. Do you wish to OVERWRITE it (Y,N)
-1520 DATA SAVE DC OPEN T(10) N0$
    : DATA SAVE DC S1$(),S3$()
    : DATA SAVE DC END
    : GOTO 40
```

40

TAEG Report No. 88

TYPE.INI

This program sets up the initializing command strings required by the typographer's subsystem. It also contains the subroutines required for addressing the computer authored text created by the AUTHOR system.

41

TYPE.INI

```
 7000 Z## Progras > AUTHOR/3  (ATH3INT1)  Version 2.0  2200 T/VP   780321
 7005 DEFFN ' 200
 7010 R4$(01)=HEX(41614262436344644565466647674868)
 7015 R4$(02)=HEX(49694A6A4B6B4C6C4D6D4F6E4F6F5070)
 7020 R4$(03)=HEX(5171527253735474557556765775787B)
 7025 R4$(04)=HEX(59795A7A20202020202020202020202O)
 7030 R4$(05)=HEX(614162426343644465456466747684B)
 7035 R4$(06)=HEX(69496A4A6B4B6C4C6D4D6E4E6F4F7050)
 7040 R4$(07)=HEX(7151725273537454755576567577858)
 7045 R4$(08)=HEX(79597A5A20202020202020202020202O)
 7050 R4$(09)=HEX(204142434445464748494A4B4C4D4E4F)
 7055 R4$(10)=HEX(20202020202020202 202020202020O)
 7060 R4$(11)=HEX(010802A7030D03E62 20202020202020)
 7065 R4$(12)=HEX(2000202000002020203020200030202O)
 7070 R4$(13)=HEX(3010311132123313341435153616371?)
 7075 R4$(14)=HEX(381839192000202020202020202020O)
 7080 R4$(15)=HEX(000102030405060708090A0B0C0D0E0F)
 7085 R4$(16)=HEX(20202020202020202020202020202O)
 7090 R4$(17)=HEX(010802A703E504E6050D202020202020)
 7095 R4$(18)=HEX(01A0010B0BCB025E0CCA03280DC70429)
 7100 R4$(19)=HEX(0EC3052A05C2062D06C407EB07C6090D)
 7105 R4$(20)=HEX(0AE5082F08CC022020202020202020O)
 7110 INIT(0A)R6$()
    : R6$(01)=HEX(01)
 7115 INIT(09)R7$()
    : R7$(01)=HEX(0D)
 7120 INIT(20)K5$()
    : K5=1
 7125 R8$(01)=HEX(400642604008121142504008113614541608114O)
 7130 R8$(02)=HEX(4006A0004021400D400A2020)
 7135 R8$(03)=HEX(FFFF2020)
 7140 R8$(04)=HEX(40071221400712214007202O)
 7145 R8$(05)=HEX(FFFF2020)
 7150 R8$(06)=HEX(A000400D1427160840OA2020)
 7155 R8$(07)=HEX(400740072020)
 7160 R8$(08)=HEX(050D14671708050A2020)
 7165 R8$(09)=HEX(05163F20556677000000)
 7170 Q0(01)=00
    : Q0(02)=00
 7175 Q0(03)=75
    : Q0(04)=06
 7180 Q0(05)=00
    : Q0(06)=00
 7185 Q0(07)=00
    : Q0(08)=00
 7190 Q0(09)=01
    : Q0(10)=1024
 7195 Q0(11)=01
    : Q0(12)=00
 7197 E8$="NOT RANGE"
 7200 RETURN

 7205 DEFFN'101(K1,K2,K3,K4)
 7210 IF K3=0THEN 7265
 7215 IF K4<>0THEN 7225
 7220 K4=81-K2
-7225 FOR K5=1TO K3
 7230 IF K5<>1THEN 7245
 7235 $GIOROW/005(A000,R8$(9))R6$()<1,K1>
 7240 GOTO 7250

-7245 $GIOROW/005(A000,R8$(9))R6$()<2,1>
-7250 $GIOCOL/005(A000,R8$(9))R7$()<1,K2>
 7255 $GIOERASE/005(A000,R8$(9))K5$()<1,K4>
 7260 NEXT K5
-7265 $GIOROW/005(A000,R8$(9))R6$()<1,K1>
 7270 $GIOCOL/005(A000,R8$(9))R7$()<1,K2>
 7275 RETURN

 7280 DEFFN'103(W1$,W1,W2,W3,W4,W5)
 7285 DEFFNW(W)=256*W+VAL(STR(W1$,2,1))
 7290 ON W5GOTO 7315,7335,7360,7380
 7295 R8$="00"
 7300 R7$="Subroutine 103"
 7305 GOSUB '000
 7310 GOTO 7455

-7315 W5=W1*10000+W2*100+W3
```

```
 7320 BIN(STR(W1$,1,1))=INT(W5/256)
 7325 BIN(STR(W1$,2,1))=W5-INT(W5/256)*256
 7330 GOTO 7455

-7335 W5=FNW(VAL(W1$))
 7340 W1=INT(W5/10000)
 7345 W2=INT((W5-(W1*10000))/100)
 7350 W3=W5-(W1*10000)-(W2*100)
 7355 GOTO 7455

-7360 GOSUB 7335
 7365 GOSUB 7380
 7370 GOSUB 7315
 7375 GOTO 7455

-7380 IF W3+W4>Q0(7)THEN 7395
 7385 W3=W3+W4
 7390 GOTO 7455

-7395 IF W2+1>Q0(6)THEN 7415
 7400 W2=W2+1
 7405 W3=W4+W3-Q0(7)
 7410 GOTO 7455

-7415 IF W1+1>Q0(5)THEN 7440
 7420 W1=W1+1
 7425 W2=1
 7430 W3=W4+W3-Q0(7)
 7435 GOTO 7455

-7440 R8$="00"
 7445 R7$="Subroutine 103"
 7450 GOSUB '000
-7455 RETURN

 7460 DEFFN'108(W6)
 7465 ON W6GOTO 7475,7475,7470,7485
-7470 PRINT "UNDEFINED BRANCH - W6 = ",W6
    : STOP
-7475 SELECT #5005
 7480 GOSUB '112
-7485 RETURN

 7490 DEFFN'112
 7495 R8$=HEX(5102)
 7500 R9$(1)="Touch RETURN"
 7505 $GIOHOME#5(4001,R8$(9))
 7510 FOR W9=1TO 16
 7515 $PACK(F=R8$)STR(R8$(9),7,2)FROMW9
 7520 W8=64
    : IF W9=1 THEN W8=63
 7525 $GIOPRINT#5(R8$(6),R8$(9))Q$()<(W9-1)*64+1.W8>
 7530 NEXT W9
 7535 GOTO 7575

 7540 FOR W9=1TO 1024STEP 64
 7545 KEYIN R8$,7550,7550
    : GOTO 7565

-7550 IF STR(R8$,1,1)<>HEX(13)THEN 7565
-7555 KEYIN R8$,7560,7560
    : GOTO 7555

-7560 IF STR(R8$,1,1)<>HEX(11)THEN 7555
-7565 W8=64
    : IF W9=1THEN W8=63
    : $GIO/005(A000400D400A,R8$(9))Q$()<W9.W8>
 7570 NEXT W9
-7575 RETURN

 7580 %## Progras > AUTHOR/3  (ATH3LST5)   Version > 2.0   2200 T/VF  780323
 7585 DEFFN'201(W6)
 7590 PRINT HEX(03)
 7595 GOSUB '101(10,1,0,0)
 7600 SELECT PRINT 005
 7605 PRINTUSING 7720," "
 7610 PRINTUSING 7725," "
 7615 SELECT PRINT 405
 7620 GOSUB '101(5,1,0,0)
 7625 PRINTUSING 7730," ";
```

```
 7630 STR(R8$(9),5,1)=" "
-7635 KEYIN R0$(1),7650,7650
 7640 $GIOWAIT/005(R8$(1),R8$(9))
 7645 GOTO 7635

-7650 IF R0$(1)>HEX(32)THEN 7660
 7655 IF R0$(1)>HEX(30)THEN 7685
-7660 GOSUB '101(5,50,0,0)
 7665 PRINT "RE-ENTER"
 7670 $GIOTONE/005(R8$(4),R8$(9))
 7675 GOSUB '101(5,50,1,0)
 7680 GOTO 7635

-7685 CONVERT R0$(1)TO W6
 7690 PRINT W6
-7695 GOSUB ' 101(7,0,0,0)
    : PRINTUSING 7735,A0$;
    : INPUT A0$
    : IF STR(A0$,,1)<>"3"AND STR(A0$,,1)<>"B" THEN 7695
    : SELECT #3 (A0$)
-7700 GOSUB ' 101(8,0,0,0)
    : PRINTUSING 7740,A1$;
    : INPUT A1$
    : IF STR(A1$,,1)<>"3"AND STR(A1$,,1)<>"B" THEN 7700
    : SELECT #1 (A1$)
 7705 GOSUB '101(16,0,0,0)
    : PRINTUSING 7745,H1$;
    : INPUT H1$
    : GOSUB '101(17,0,0,0)
    : PRINTUSING 7750,H2$;
    : INPUT H2$
    : GOSUB '101(18,0,0,0)
    : PRINTUSING 7755,H3$;
    : INPUT H3$
    : GOSUB '101(19,0,0,0)
    : PRINTUSING 7760,H4$;
    : INPUT H4$
 7710 GOSUB '101(20,0,0,0)
    : PRINTUSING 7765,H5$;
    : INPUT H5$
    : GOSUB '101(21,0,0,0)
    : PRINTUSING 7770,H6$(1),H6$(2),H6$(3):
    : INPUT H6$(1),H6$(2),H6$(3)
 7715 GOSUB ' 101(16,1,7,0)
-7720 % Output Selection     #     1 - CRT  (WAIT AFTER EACH PAGE)
-7725 %                      #     2 - CRT  (NO WAIT AFTER EACH PAGE)
-7730 % Enter Option From Output Selection Menu > #
-7735 % Enter Disk Address AUTHOR module Diskette (###)    >
-7740 % Enter Disk Address POSTPROC Output Diskette (###) >
-7745 % Enter Job Name (####################################):
-7750 % Enter P/O Number (##########):
-7755 % Enter Todays Date (##########):
-7760 % Enter Operator's Initials (###):
-7765 % Enter Typographic's P/O (####):
-7770 % Enter Job  sequence number, first . last (####. ####. ####):
 7775 RETURN

 7780 DEFFN'202(W6)
 7785 IF W6=3THEN 7965
 7790 INIT(20)Q$()
 7795 INIT(20)Q1$()
 7800 D1=0
 7805 D2=128
 7810 DATA LOAD BA T#3,(0,D9)Q1$()
 7815 ROTATE(STR(Q1$(25),1,1),4)
 7820 D6=VAL(STR(Q1$(25),1,1))
 7825 DATA LOAD BA T#3,(1,D9)Q1$()
 7830 REM                                  : initialize output file

 7835  L2$()=" "
 7840 DATA LOAD DC OPEN T #1,"TYPESET"
 7845 LIMITS T #1, 01,02,03
 7865 REM                                  : define header and fonts used
 7870 INIT(20)L2$()
    : L2$()=HEX(7B) & "do20" & HEX(7D)
    : GOSUB '100("mx531,2,3,084,555,086,534.392.393.394.395.396.541")
    : GOSUB '97
    : GOTO 7880

 7875 DEFFN'97
```

```
     : GOSUB '100("af401")
     : GOSUB '98("WP")
     : GOSUB '98(H1$)
     : GOSUB '98(H2$)
     : GOSUB '98(H3$)
     : GOSUB '98(H4$)
     : GOSUB '98(H5$)
     : GOSUB '98(H6$(1))
     : GOSUB '98(H6$(2))
     : GOSUB '98(H6$(3))
     : GOSUB '100("d1+")
     : RETURN

-7880 REM                                   : 1- body text - cntry schbk (084) -
      12/16
 7885 GOSUB '100("df1")
    : GOSUB '100("cf4,12,16")
    : GOSUB '100("ef")
 7890 REM                                   : 2- bold text - cntry schbk bold (08
      6) - 12/16
 7895 GOSUB '100("df2")
    : GOSUB '100("cf6,12,16")
    : GOSUB '100("ef")
 7900 REM                                   : 3- headings-helvetica bold (396)-14
      /17
 7905 GOSUB '100("df3")
    : GOSUB '100("cf12,14,17")
    : GOSUB '100("ef")
 7910 REM                                   : 4- page no's helvetica (394)-14/16
 7915 GOSUB '100("df4")
    : GOSUB '100("cf10,14,16")
    : GOSUB '100("ef")
 7920 REM                                   : 5- titles -  helvetica bold (396)-1
      8/24
 7925 GOSUB '100("df5")
    : GOSUB '100("cf12,18,24")
    : GOSUB '100("ef")
 7930 REM                                   : 6- subtitles helvetica (394)-14/17
 7935 GOSUB '100("df6")
    : GOSUB '100("cf10,14,17")
    : GOSUB '100("ef")
 7940 REM                                   : 7- pub/org - helvetica BOLD  (396)-
       12/14
 7945 GOSUB '100("df7")
    : GOSUB '100("cf12,12,14")
    : GOSUB '100("ef")
 7950 REM                                   : 8- symbols - pi-font     -  16/16
 7955 GOSUB '100("df8")
    :  GOSUB '100("cf5,16,16")
    : GOSUB '100("ef")
 7960 IF W6<>1THEN 8080
-7965 W6=1
 7970 SELECT PRINT 005
 7975 PRINT HEX(03)
 7980 RESTORE 1
 7985 FOR W9=1TO 13
 7990 READ R9$(1)
 7995 PRINTUSING 8010,R9$(1)
 8000 NEXT W9
 8005 W6=1
-8010 % ############################################################
 8015 DATA "Instructions for Referencing pages:"
 8020 DATA " - After each page is displayed a prompt will appear in"
 8025 DATA "   the lower left corner of screen."
 8030 DATA " - Once you have entered the next page number touch"
 8035 DATA "   the 'RETURN' key."
 8040 DATA " - To perform special functions you may enter:"
 8045 DATA "     START - Begin displaying Learning Module."
 8050 DATA "     NEXT  - Go to next sequintal page."
 8055 DATA "     STOP  - Stop listing."
 8060 DATA "     "
 8065 DATA "     LIST  - List the entire learning module."
 8070 DATA "     "
 8075 DATA "     HELP  - This text."
-8080 RETURN

-8085 DEFFN'203
-8090 GOSUB '205(W6,W3,D7$)
 8095 IF D7$<>"STOP"THEN 8150
 8100    Q$(1)="end of file"
```

```
 8105    GOTO  8275

 8110 IF W4$<>"TAPE"THEN 8115
-8115 PRINT HEX(03)
 8120 SELECT PRINT 005
 8125 PRINTUSING 8260," "
 8130 PRINTUSING 8265," "
 8135 PRINTUSING 8270," "
 8140 GOSUB '101(12,1,0,0)
 8145 END

-8150 IF D7$<>"PRIN"THEN 8160
 8155 D7$="ERRO"
-8160 IF D7$<>"ERRO"THEN 8190
 8165 GOSUB '101(17,9,0,0)
 8170 PRINT "RE-ENTER"
 8175 $GIOTONE/005(R8$(4),R8$(9))
 8180 GOSUB '101(17,9,1,10)
 8185 GOTO 8090

-8190 IF D7$<>"HELP"THEN 8205
 8195 GOSUB '202(3)
 8200 GOTO 8090

-8205 IF D7$<>"EDIT"THEN 8215
 8210 GOTO 8255

-8215 IF D7$<>"LIST"THEN 8255
 8220 W6=0
 8225 W2$=" "
    : W3$=" "
 8230 W4$=" "
    : W5$=" "
 8235 W6$=" "
 8240 GOSUB '201(W6)
 8245 GOSUB '202(W6)
 8250 GOTO 8085

-8255 GOSUB '207(D7$)
-8260 % AUTHOR;  An Automatic Authoring System #
-8265 % Phase III   Construction Processing    #
-8270 % L I S T I N G    C O M P L E T E D     #
-8275 RETURN

 8280 DEFFN'205(W6,W3,D7$)
 8285 IF W6=01THEN 8345
-8290 D2=D2+2
 8295 IF D2<128THEN 8325
 8300 D1=D1+1
 8305 IF D1<17THEN 8315
 8310 GOSUB '000
-8315 DATA LOAD BA T#3,(D1,D9)Q1$()
 8320 D2=1
-8325 D7$=Q1$(D2+1)
 8330 IF Q1$(D2)<>HEX(FFFF)THEN 8430
 8335 D7$="STOP"
 8340 GOTO 8430

-8345 GOSUB '101(17,1,1,20)
 8350 PRINT HEX(20203F3F3F3F080808080808):
 8355 D7$="NEXT"
 8357 IF E8$<>"RANGE" THEN 8360
    : IF STR(Q$(1),59,4) <>E7$ THEN 8365
    : E8$="NOT RANGE"
-8360 INPUT D7$
-8365 PRINT "System Working...";
 8370 AND (D7$,DF)
 8375 $TRAN(D7$,R4%()<193,24>)R
 8380 IF D7$="STOP"THEN 8430
 8385 IF D7$="LIST"THEN 8430
 8390 IF D7$="HELP"THEN 8430
 8395 IF D7$="NEXT"THEN 8290
 8396 IF D7$<>"RANG" THEN 8400
    : INPUT "ENTER START AND END",D7$,E7$
    : E8$="RANGE"
    : GOTO 8365

-8400 IF D7$<>"STAR"THEN 8420
 8405 D1=0
 8410 D2=129
```

```
 8415 GOTO 8290

-8420 GOSUB '209(D7$)
 8425 GOSUB '206(D7$,D1,D2)
-8430 RETURN

 8435 DEFFN'206(D7$,D1,D2)
 8440 DEFFNR(R)=256*R+VAL(STR(R2$(1),2,1))
 8445 IF D7$="ERRO"THEN 8545
 8450 D3=D1
 8455 D4=D2
 8460 MAT SEARCHQ1$(),=STR(D7$,1,2)TO R2$()STEP 4
 8465 IF R2$(1)<>HEX(0000)THEN 8530
 8470 D3=0
 8475 FOR D5=1TO 10
 8480 DATA LOAD BA T#3,(D5,D9)Q1$()
 8485 MAT SEARCHQ1$(),=STR(D7$,1,2)TO R2$()STEP 4
 8490 IF R2$(1)=HEX(0000)THEN 8505
 8495 D3=D5
 8500 D5=10
-8505 NEXT D5
 8510 IF D3<>0THEN 8530
 8515 D7$="ERRO"
 8520 DATA LOAD BA T#3,(D1,D9)Q1$()
 8525 GOTO 8545

-8530 D1=D3
 8535 D2=INT(FNR(VAL(R2$(1)))/2)+1
 8540 D7$=Q1$(D2+1)
-8545 RETURN

 8550 DEFFN'207(D7$)
 8555 IF W3=99THEN 8565
 8560 INIT(20)Q$()
-8565 GOSUB '211(D7$)
 8570 FOR W9=1TO 1024STEP 256
 8575 DATA LOAD BA T#3,(D7$,D7$)Q9$()
 8580 MAT COPY Q9$()<1,256>TO Q$()<W9,256>
 8585 NEXT W9
 8590 RETURN

 8595 DEFFN'209(D7$)
 8600 D8=7
 8605 D9=14
 8610 IF STR(D7$,2,1)>HEX(40)THEN 8655
 8615 IF NUM(D7$)=4THEN 8630
 8620 D7$="ERRO"
 8625 GOTO 8715

-8630 CONVERT D7$TO D7
 8635 W1=INT(D7/(D8*D9))+1
 8640 W2=INT(((D7-(W1-1)*(D8*D9))-1)/D9)+1
 8645 W3=D7-((W1-1)*(D8*D9)+(W2-1)*D9)
 8650 GOTO 8705

-8655 $TRAN(D7$<2,1>,R4$()<225,18>)0F
 8660 IF NUM(STR(D7$,1,1))<>1THEN 8680
 8665 IF NUM(STR(D7$,3,2))<>2THEN 8680
 8670 IF STR(D7$,2,1)>HEX(0F)THEN 8680
 8675 IF STR(D7$,2,1)>HEX(00)THEN 8690
-8680 D7$="ERRO"
 8685 GOTO 8715

-8690 CONVERT STR(D7$,1,1)TO W1
 8695 CONVERT STR(D7$,3,2)TO W3
 8700 W2=VAL(STR(D7$,2,1))
-8705 GOSUB '103(W1$,W1,W2,W3,0,1)
 8710 D7$=W1$
-8715 RETURN

 8720 DEFFN'210(W6)
 8725 GOSUB '108(W6)
 8730 RETURN

 8735 DEFFN'211(D7$)
 8740 MAT REDIM Q9$(64)4
 8745 STR(D7$,3,2)=STR(D7$,1,2)
 8750 AND (STR(D7$,1,1),0F)
 8755 AND (STR(D7$,3,1),F0)
 8760 ROTATE(STR(D7$,3,1),4)
```


47

```
 8765 W9=VAL(STR(D7$,3,1))
-8770 IF W9=D6THEN 8825
 8775 GOSUB '101(17,1,2,0)
 8780 PRINTUSING 8820,W9
-8785 KEYIN R0$(1),8800,8800
 8790 $GIOPAUSE/005(R8$(4),R8$(9))
 8795 GOTO 8785

-8800 DATA LOAD BA T#3,(0,D9)Q9$()
 8805 ROTATE(STR(Q9$(13),1,1),4)
 8810 D6=VAL(STR(Q9$(13),1,1))
 8815 GOTO 8770

-8820 % MOUNT Volume > ## Output Disk ...  'RETURN' to Continue
-8825 RETURN
```

TAEG Report No. 88

TYPE.TST

This program contains the instructions necessary to substitute the required typesetting command strings for the mnemonic codes used by the AUTHOR system for special symbols, inserts the other typesetting commands required for setting the remainder of the text, and stores the resulting composed text in running text stream form ready for transmission to the typographer. It does [illegible]is using the special hexadecimal digit code placed in the last column [illegible]tion on the first line of each page or frame by the AUTHOR system. This [illegible] directs the program to the appropriate subroutine for setting that page [illegible] frame in the specified format.

TYPE.TST

```
-0000 % # UNDEFINE PROGRAM REFERENCE (23 October /9)
 0010 %## pgm: AUTHOR System V3.0  (TYPE.TST) read and convert module (full/page
      )
 0020 COM S1$(128)4, S3$(128,3)13
 0030 COM A0$3,A1$3
 0040 DIM R4$(20)16
 0050 DIM R8$(10)24,R9$(1)64
 0060 DIM B0
 0070 DIM D0,D1,D2,D3,D4,D5,D6,D7,D8,D9
 0080 DIM D4$4,D7$4
 0090 DIM K1,K2,K3,K4,K5$(80)1,R6$(24)1,R7$(80)1
 0100 DIM W1,W2,W3,W4,W5,W6,W8,W9,W1$
 0110 DIM W2$16,W3$16,W4$16,W5$16,W6$16
 0120 DIM R1,R2,R4,R5,R6,R8,R9
 0130 DIM R8$2
 0140 DIM R0$(1)1,R2$(2)2
 0150 DIM Q0(12)
 0160 DIM Q1$(128)2
 0170 DIM Q9$(64)4
 0180 REM                                    : variable definition/initialization

 0190  DIM F
    : REM FONT INVOKED
 0200  DIM K$64
    : REM COMMAND BUFFER
 0210  DIM L
    : REM LINE ADDRESSED
 0220  DIM L1$(64)1
    : REM LINE INPUT
 0230  DIM L2$(5)64,L3$(5)64
    : REM LINE OUTPUT BUFFER
 0240  DIM  M0
    : REM DEFAULT MODE
 0250  DIM M1
    : REM CURRENT MODE
 0260  DIM M2
    : REM NEW MODE
 0270  DIM M$1
    : REM  MODE IN EFFECT
 0280  DIM O$(4)64
    : REM DISK OUTPUT BUFFER
 0290  DIM P
    : REM POINTS TO BE LEADED
 0300  DIM P$2
    : REM POINTS LEADING COMMAND
 0310  DIM Q$(16)64
    : REM READ-IN BUFFER
 0320  DIM S
    : REM SYMBOL INPUT LENGTH
 0330  DIM S2
    : REM SYMBOL INPUT-OUTPUT ADDRESS
 0340  DIM S$124
    : REM SEARCH POINTER TBL
 0350  DIM T$5
    : REM  TERMINATION
 0360  DIM V
    : REM VERTICAL MEASURE
 0370  DIM W$64
    : REM WORD
 0380  DIM W$(128)2
    : REM WORD MAP
 0390  DIM X0$6
    : REM XNLT TABLE - FORMAT
 0400       X0$=HEX(010E030F2020)
 0410 DIM X1$16
    : REM XNLT TABLE - INPUT
 0420       X1$=HEX(802380A3B120B1A02525A5A5/EFE2020)
 0430 DIM X2$4
    : REM XNLT TABLE - OUTPUT
 0440       X2$=HEX(207F2020)
 0450 DIM X3$4
    : REM XNLT TABLE - UNDERLINING
 0460       X3$=HEX(205F2020)
 0470 REM                                    : header information
    : DIM H1$30, H2$10, H3$10, H4$3, H5$4, H6$(3)4, H$64
    : H1$="U.S. NAVY (TAEG) PKG 1          "
    : H2$="4713-1P037"
    : H3$="dd-mmm-yy"
    : H4$="f1k"
    : H5$="0990"
    : H6$(1)="0001"
    : H6$(2)="0001"
    : H6$(3)="0001"
 0480 %##            END OF COMMON STORAGE AREA          ##
 0490 %##       END OF COOMON STORAGE AREA      ##
 0500 REM %+
```

Abstract

```
0510 I      This program will process all pages output by the AUTHOR
0520 I      System and add typesetting code which will allow those pages
0530 I      to be transmitted to a typesetter for final output.
0540 REM              :begin main processing loop
0550 GOSUB '200
0560 GOSUB '201(W6)
0570 GOSUB '202(W6)
0580 REM              :init total page output
   : O8=0
   : V8=99
   : V9=99
0590 REM              ; invoke page # mode
   : GOSUB '102(4)
0600 REM              ; start next page if error
   : REM ON ERROR E8$,E9$ GOTO 570
-0610 REM             ; set cut marks
   : GOSUB '99("+ +")
   : GOSUB '99(HEX(05))
0620 GOSUB '203
0630 GOSUB '210(W6)
0640 REM              ; test for end-of-file
   : IF       Q$(1)<>"end of file" THEN 740
   : REM THEN 780
0650 REM              ; save end-of-file info
0660                   GOSUB '100("te")
   : GOSUB '99(" $te")
   : L2$()=L2$() & ALL(00)
0670                          GOSUB '99(" ")
0680       INIT (00) L2$()
   : GOSUB '99(" ")
   : INIT (00) L2$()
   : GOSUB '99(" ")
0690       DBACKSPACE #1,BEG
0700       DSKIP #1,(O3-O1)S
0710       DATA SAVE DC #1,END
0720       PRINT "COMPOSITION COMPLETE"
0730       STOP
-0740 REM         ;incr total pages set
   : O8= O8 +1
0750 REM          ; test for end of N-pages)
   : IF     O8-INT((O8-1)/4)*4 <>1 OR O8 = 1  THEN  770
   : IF O8-INT((O8-1)/12)*12<>1THEN 751
   : GOSUB '99(HEX(07))
   : GOSUB '100("do20")
   : GOSUB '97
   : GOSUB '100("g4")
   : GOTO 752

-0751 FOR   I=1TO 3
   : GOSUB '100("os")
   : NEXT  I
-0752 REM    set new cut marks
   : GOSUB '99("+ +")
   : GOSUB '99(HEX(05))
0760 REM          ; set top of page
-0770       GOSUB '100("e112")
   : IF STR(Q$(1),59,1)="1"THEN GOSUB '99(STR(Q$(1).60.3))
   : ELSE GOSUB '99(STR(Q$(1),59,4))
   : GOSUB '100("qc")
0780 REM    ; set according to format code:SELECTPRINT215(132):PRINT Q$(1).HEXOF
     (STR(Q$(1),64,1)),VAL(STR(Q$(1),64,1)):GOTO 620
0790  ON VAL(STR(Q$(1),64))-00 GOSUB   870.1100.1100.1100.0.0.1140.1260.1360.11
   34,1131,1190,0,0,0,0
   : ELSE GOTO 800
   :      GOTO 820

-0800  ON VAL(STR(Q$(1),64))-16 GOSUB      0.   0.1132.1440.0.0.1100.1590.1100.0
     ,1710,1360,1360,1135,0,2000
   : ELSE GOTO 810
   :      GOTO 820

-0810  ON VAL(STR(Q$(1),64))-32 GOSUB   1860.1930.1100.1100.1133.1260.1360.1150.
     1100,1100,1136,0,1100,0,0,0
   : ELSE STOP
   :      GOTO 820

-0820 REM         ; add lead above bottom page no.

   : GOSUB '111(16)
0830 REM          ; invoke page no. mode
   : GOSUB '102(4)
0840 REM          ; set bottom page no.
   : IF STR(Q$(1),59,1)="1"THEN GOSUB '99(STR(Q$(1).60.3))
   : ELSE GOSUB '99(STR(Q$(1),59,4))
   : GOSUB '100("qc")
   : GOSUB '100("e112")
0850 REM          ; continue processing loop
   : GOTO 610

0860 REM Z↑
```

TITLE PAGE FORMAT (pp 1)

```
-0870 REM          ; invoke vert. just.
   : GOSUB '100("vm55p8")
 0880 REM          ;add leading above title
   : GOSUB '100("vb")
   : GOSUB '100("vb")
 0890 REM          ; invoke title mode
   : M0=5
 0900 REM          ; init line index
   : L=1
 0910 REM          ; set title
   : GOSUB '121(99)
 0920 REM          ; add leading above title
   : GOSUB '100("vb")
 0930 REM          ; invoke subtitle mode
   : M0=6
 0940 REM          ; set subtitle
   : GOSUB '121(99)
 0950 REM          ; add leading above authors
   : GOSUB '100("vb")
 0960 REM          ; invoke author mode
   : M0=1
 0970 REM          ;  set authors
   : GOSUB '121(99)
 0980 REM          ; add leading above date
   : GOSUB '100("vb")
 0990 REM          ; invoke date mode
   : M0=4
 1000 REM          ; set date/ quad ctr
   : GOSUB '121(99)
 1005 REM          ; set credits
   : M0=2
   : GOSUB '100("vb")
   : GOSUB '121(99)
   : GOSUB '100("vb")
 1010 REM          ; add leading above org/pub
   : GOSUB '100("vb")
   : GOSUB '100("vb")
 1020 REM          ; invoke org/pub mode
   : M0=7
 1030 REM          ;  set org/pub
   : GOSUB '121(99)
   : GOSUB '100("vb")
   : GOSUB '100("cf4,6,6")
   : GOSUB '99("TYPOGRAPHY by F. Laurence Keeler. Ph.D.")
   : GOSUB '100("qc")
 1040 REM          ; add leading above GOTO
   : GOSUB '100("vb")
 1050 REM          ; invoke GOTO     mode
   : M0=1
 1060 REM          ; set goto / quad right
   : GOSUB '130(Q$(16))
   : GOSUB '100("is6")
   : GOSUB '100("qr")
 1070 GOSUB '100("vb")
   : GOSUB '99(HEX(02))
   : GOSUB '100("o10")
   : GOSUB '100("vj")
 1080 RETURN

 1090 REM %↑
```

```
        simple 2-col format

-1100 REM       ; set heading
    : GOSUB '122
-1110 REM       ; define tabs
    : GOSUB '100("dt6g,10p,2g,21p,6p")
 1120 REM       ; set remaining lines
    : FOR L =2 TO 16
    :     GOSUB '123
    : NEXT L
 1130 RETURN

-1131 IF STR(Q$(10),59,1) <> "," THEN 1100
    : STR(Q$(10),59,1) = " "
    : STR(Q$(11),24,41) = "and bottom of each page."
    : STR(Q$(14),24,41) = "lower right corner."
    : GOTO 1100

-1132 IF STR(Q$(3),27,4)="at D" THEN STR(Q$(3).27.38)="at the DEFINITION and MEM
      ORY AID for"
    : IF STR(Q$(5),46,5)="and D" THEN STR(Q$(5),46,19)="and the"
    : IF STR(Q$(6),27,4)="to y"THEN STR(Q$(6).27.38)="DEFINITION to yourself as
      you look at"
    : GOTO 1100

-1133 IF STR(Q$(16),35,16)="For test symbols" THEN STR(Q$(16).35.16)=" "
    : GOTO 1100

-1134 FOR L=2TO 12
    : L1$()=Q$(L)
    : IF L1$(26)<>" "THEN STR(Q$(L),24,41)=STR(L1$().21.41)
    : NEXT L
    : GOTO 1100

-1135 IF STR(Q$(10),28,4)="To P"THEN STR(Q$(10).28.4)="To p"
    : GOTO 1100

-1136 IF STR(Q$(9),27,12) = "training.  Y" THEN STR(Q$(9).27.38)="training. Your
       memory plays tricks"
    : GOTO 1100

-1140 REM %↑
```

optional criterion test

```
    : GOSUB '122
    : GOSUB '102(2)
    : M0=2
    : FOR L=2TO 4
    : GOSUB '100("bt")
    : GOSUB '130(STR(Q$(L),6,16))
    : GOSUB '100("ql")
    : GOSUB '100("et")
    : STR(Q$(L),6,16)=" "
    : V8=V8-16
    : NEXT L
    : GOSUB '100("dt6g,12p,21p,6p")
    : GOSUB '100("bt")
    : GOSUB '100("tl")
    : GOSUB '130(Q$(2))
    : GOSUB '100("tc")
    : GOSUB '100("et")
    : FOR L=3TO 16
    : GOSUB '123
    : NEXT L
    : RETURN

1149 REM Z↑
```

```
        criterion test ##

-1150 REM         ; set heading
    : GOSUB '122
 1160 REM         ; define tabs
    : GOSUB "100("dt6g,10p,2g,21p,6p")
 1170 REM         ; set remaining lines
    : FOR L =2 TO 15
    :     GOSUB '126
    : NEXT L
    : GOSUB '129
 1180 RETURN

-1190 REM %↑
```

symbols in set

```
1200 REM       : set heading
   : GOSUB '122
1210 REM       : set line 2
   : GOSUB '100("dt18g,21p,6p")
   : M0=1
   : GOSUB '100("bt")
   : GOSUB '130(STR(Q$(2),24,41))
   : GOSUB '100("tl")
   : GOSUB '100("et")
1215 REM       ;test for graphic symbols
   : MAT SEARCH Q$(4),="Z" TO W$()
   : IF VAL(W$(1),2)<>0 THEN 1220
   : GOSUB '111(16)
   : GOSUB '100("du3")
   : FOR L=4 TO 6
   : GOSUB '100("is18p")
   : GOSUB '130(STR(Q$(L),24,41))
   : GOSUB '100("is6p")
   : GOSUB '100("jl")
   : NEXT L
   : GOSUB '100("du2")
   : GOTO 1230

-1220 REM      ;set symbols
   : GOSUB '111(16)
   : GOSUB '100("dt6g,12p,4p,4p,4p,4p,4p,4p,/p")
   : Q$(5) = Q$(5) & STR(Q$(6),24)
   : FOR L=4 TO 5
   : GOSUB '139
   : NEXT L
   : GOSUB '111(16)
-1230 REM      ;set remaining text
   : GOSUB '100("dt6g,12p,21p,6p")
   : FOR L=7 TO 16
   : GOSUB '123
   : NEXT L
   : RETURN

1240 REM Z↑
```

criterion test - symbols

```
1250 REM       ; (pp 10,51)
-1260 REM      ; set heading
   : GOSUB '122
1270 REM       ;invoke text mode
   : M0=1
1280 REM       ; define tabs
   : GOSUB '100("dt6g,3p,1g,10p,1g,10p,1q.3p.1q.3p.1q.8p.6p")
1290 FOR L= 2 TO 15
   : GOSUB '111(7)
   : V8=V8-7
1300 GOSUB '124
1310 NEXT L
1320 REM       ; define tabs
   : GOSUB '100("dt6g,10p,2g,21p,6p")
   : REM       ; set goto ref.
   : L=16
   : GOSUB '123
1330 RETURN

1340 REM %↑
```


57

```
     criterion test - answers

 1350 REM      ; (pp 11,12,52)
-1360 REM       ; set heading
    : GOSUB '122
 1370 T=0
 1380 FOR L=2 TO 16
    : IF STR(Q$(L),6,1) > HEX(A0) THEN 1400
    : IF STR(Q$(L),8,1) = "." THEN 1410
    : IF  T <> 1 THEN GOSUB '100("dt6g,10p.2q.21p.6p")
    : T=1
    : GOSUB '123
-1390 NEXT L
    : RETURN

-1400 REM     ;set col hdgs
    : IF T <> 2 THEN GOSUB '100("dt6g,3p.10p.20p.6p")
    : T = 2
    : AND (Q$(L),7F)
    : GOSUB '100("bt")
    : M0 = 2
    : GOSUB '100("nt")
    : GOSUB '130(STR(Q$(L),1,24))
    : GOSUB '100("tc")
    : GOSUB '130(STR(Q$(L),26,30))
    : GOSUB '100("tl")
    : GOSUB '100("et")
    : GOTO 1390

-1410 REM     ;set # - sym - ans
    : IF T <> 2 THEN GOSUB '100("dt6g,3p.10p.20p,6p")
    : T = 2
    : GOSUB '111(7)
    : V8=V8-7
    : M0=1
    : GOSUB '127
    : GOTO 1390

 1420 REM %↑
```

58

learn symbol definitions

```
 1430 REM      ;(pp 18)
-1440 REM      ; set heading
    : GOSUB '122
 1450 REM      ; define 3-col. tabs
    : GOSUB '100("dt6g,11p,11p,11p,6p")
 1460 REM      ; set col hdgs
    : IF Q$(2)<>" " THEN 1470
    : GOSUB '111(16)
    : GOTO 1520

-1470 GOSUB '100("bt")
    : M0=2
    : AND (Q$(2),7F)
 1480 REM      ; set col 1 (definition) heading
    : GOSUB '130(STR(Q$(2),1,25))
    : GOSUB '100("tr")
 1490 REM      ; set col 2 (symbol) heading
    : GOSUB '130(STR(Q$(2),27,15))
    : GOSUB '100("tc")
 1500 REM      ; set col 3 (mem aid) heading
    : GOSUB '130(STR(Q$(2),43,20))
    : GOSUB '100("tl")
 1510 GOSUB '100("et")
    : M0=1
-1520 FOR L=3 TO 15
 1530 GOSUB '125
 1540 NEXT L
 1550 REM      ; define tabs
    : GOSUB '100("dt6g,10p,2g,21p,6p")
    : REM      ; set goto ref
    : GOSUB '129
 1560 RETURN

 1570 REM %↑
```

```
    practice symbol definitions

-1590 REM      ; set heading
    : GOSUB '122
 1595 REM      ;test for graphic symbols
    : MAT SEARCHQ$(2),="I"TO W$()
    : IF VAL(W$(1),2)<>0THEN 1600
    : GOSUB '100("du3")
    : FOR L=2TO 6
    : IF STR(Q$(L),6,16)=" "THEN 1596
    : M0=2
    : GOSUB '100("is6p")
    : GOSUB '130(STR(Q$(L),6,16))
    : GOSUB '100("ql")
    : GOSUB '100("ol0")
-1596 GOSUB '100("is18p")
    : M0=1
    : GOSUB '130(STR(Q$(L),24,41))
    : GOSUB '100("is6p")
    : GOSUB '100("jl")
    : NEXT L
    : GOTO 1620

-1600 REM      ; define tabs
    : GOSUB '100("dt6g,12p,4p,4p,4p,4p,4p,7p")
 1601 SELECT PRINT 215(132)
    : FOR L=1TO 6
    : PRINT "INITIAL    ",Q$(L)
    : NEXT L
    : PRINT
-1602 FOR L=3TO 6
-1603 FOR I=1TO 4
    : IF STR(Q$(L),17+7*I,6)<>STR(Q$(L-1),17+7*I,6)THEN 1604
    : STR(Q$(L),17+7*I,6)=STR(Q$(L),17+7*I+7,6)
    : STR(Q$(L),17+7*I+7,6)=STR(Q$(L-1),17+7*I,6)
-1604 NEXT I
    : IF STR(Q$(L),52,6)<>STR(Q$(L-1),52,6)THEN 1606
    : STR(Q$(L),52,6)=STR(Q$(L),24,6)
    : STR(Q$(L),24,6)=STR(Q$(L-1),52,6)
    : GOTO 1603

-1606 NEXT L
    : FOR I=2TO 5
 1607 FOR L=2TO 5
    : IF STR(Q$(L),17+7*I,6)<>STR(Q$(L),17+7*(I-1),6)THEN 1608
    : STR(Q$(L),17+7*I,6)=STR(Q$(L+1),17+7*I,6)
    : STR(Q$(L+1),17+7*I,6)=STR(Q$(L),17+7*(I-1),6)
-1608 NEXT L
    : IF STR(Q$(6),17+7*I,6)<>STR(Q$(6),17+7*(I-1),6)THEN 1609
    : STR(Q$(6),17+7*I,6)=STR(Q$(2),17+I,6)
    : STR(Q$(2),17+7*I,6)=STR(Q$(6),17+7*(I-1),6)
    : GOTO 1602

-1609 NEXT I
    : FOR L=1TO 6
    : PRINT "FINAL      ",Q$(L)
    : NEXT L
    : PRINT
    : PRINT
    : SELECT PRINT 005(80)
 1610 FOR L = 2 TO 6
    : GOSUB '139
    : NEXT L
-1620 REM      ;set rule
    : GOSUB '100("dt6g,33p,6p")
    : GOSUB '100("bt")
    : GOSUB '102(1)
    : GOSUB '100("li")
    : GOSUB '100("tj")
    : GOSUB '100("et")
 1630 REM      ; set col hdgs
    : AND (Q$(8),7F)
    : GOSUB '100("dt6g,10p,2g,21p,6p")
    : GOSUB '100("bt")
    : M0=2
    : GOSUB '130(STR(Q$(8),1,20))
    : GOSUB '100("tc")
    : GOSUB '130(STR(Q$(8),22,43))
    : GOSUB '100("tl")

    : GOSUB '100("et")
    : M0=1
 1640 FOR L = 9 TO 15
    : GOSUB '111(4)
    : V8=V8-4
 1650 GOSUB '128(24)
 1660 NEXT L
 1670 REM      ; define tabs
    : REM      ; set goto ref
    : GOSUB '129
 1680 RETURN
```

60

test yourself

```
 1700 REM      ;test yourself
-1710 REM      ; set heading
    : GOSUB '122
 1720 REM      ; define tabs
    : GOSUB '100("dt6g,10p,2g,21p,6p")
 1730 REM      ; set directions
    : FOR L=2 TO 9
    : GOSUB '123
    : NEXT L
 1740 REM      ; define tabs
    : GOSUB '100("dt6g,10p,3p,1g,10p,3p,1g,10p,5p")
 1750 REM      ; set test items
    : FOR L = 10 TO 15
 1760 REM      ; compute start of text
    : L1=POS(Q$(L)<>" ")
    : REM      ; not a blank line
    : IF L1<>0 THEN 1770
    : GOSUB '111(16)
    : GOTO 1830

-1770 REM      ; add leading & begin tab
    : GOSUB '111(4)
    : V8=V8-4
    : GOSUB '100("bt")
    : REM      ;if no side heading
    : IF L1> 23 THEN 1780
    : M0=2
    : GOSUB '130(STR(Q$(L),6,18))
-1780 REM      ; tab/invoke text mode
    : GOSUB '100("tl")
 1790 REM      ; set item number for col [1]
    : M0=1
    : GOSUB '130(STR(Q$(L),24,3))
    : GOSUB '100("tr")
 1800 REM      ; set symbol  for col [1]
    : GOSUB '130(STR(Q$(L),28,15))
    : GOSUB '100("tl")
 1810 REM      ; set item number for col [2]
    : M0=1
    : GOSUB '130(STR(Q$(L),44,3))
    : GOSUB '100("tr")
 1820 REM      ; set symbol  for col [2]
    : GOSUB '130(STR(Q$(L),48,15))
    : GOSUB '100("tl")
    : GOSUB '100("et")
-1830 NEXT L
 1840 REM      ; define tabs
    : GOSUB '100("dt6g,10p,2g,21p,6p")
    : REM      ;set goto ref
    : GOSUB '129
 1850 RETURN

 1860 REM %↑
```

practice all symbols

```
 1870 REM      ;set head
    : GOSUB '122
 1875 REM      ;test for graphic symbols
    : MAT SEARCHQ$(2),="%"TO W$()
    : IF VAL(W$(1),2)<>0THEN 1880
    : GOSUB '100("du3")
    : FOR L=2TO 15
    : IF STR(Q$(L),6,16)=" "THEN 1876
    : M0=2
    : GOSUB '100("is6p")
    : GOSUB '130(STR(Q$(L),6,16))
    : GOSUB '100("ql")
    : GOSUB '100("ol0")
-1876 GOSUB '100("is18p")
    : M0=1
    : GOSUB '130(STR(Q$(L),24,41))
    : GOSUB '100("is6p")
    : GOSUB '100("jl")
    : NEXT L
    : GOSUB '100("du2")
    : GOTO 1900

-1880 REM      ;define tabs
    : GOSUB '100("dt6g,10p,2g,4p,4p,4p,4p.4p.7p")
 1881 SELECT PRINT 215(132)
    : FOR L=1TO 15
    : PRINT "INITIAL    ",Q$(L)
    : NEXT L
    : PRINT
-1882 FOR L=3TO 15
-1883 FOR I=1TO 4
    : IF STR(Q$(L),17 +7*I,6)<>S1R(Q$(L-1).17+7*I.6)THEN 1884
    : STR(Q$(L),17+7*I,6)=STR(Q$(L),17+7*I+7.6)
    : STR(Q$(L),17+7*I+7,6)=STR(Q$(L-1),17+7*I.6)
-1884 NEXT I
    : IF STR(Q$(L),52,6)<>STR(Q$(L-1),52.6)THEN 1886
    : STR(Q$(L),52,6)=STR(Q$(L),24,6)
    : STR(Q$(L),24,6)=STR(Q$(L-1),52,6)
    : GOTO 1883

-1886 NEXT L
    : FOR I=2TO 5
 1887 FOR L=2TO 14
    : IF STR(Q$(L),17+7*I,6)<>STR(Q$(L),17+7*(I-1),6)THEN 1888
    : STR(Q$(L),17+7*I,6)=STR(Q$(L+1),17+7*I.6)
    : STR(Q$(L+1),17+7*I,6)=STR(Q$(L),17+7*(I-1).6)
-1888 NEXT L
    : IF STR(Q$(15),17+7*I,6)<>STR(Q$(15).17+7*(I-1).6)THEN 1889
    : STR(Q$(15),17+7*I,6)=STR(Q$(2),17+I.6)
    : STR(Q$(2),17+7*I,6)=STR(Q$(15),17+7*(I-1).6)
    : GOTO 1882

-1889 NEXT I
    : FOR L=1TO 15
    : PRINT "FINAL      ",Q$(L)
    : NEXT L
    : PRINT
    : PRINT
    : SELECT PRINT 005(80)
 1890 REM      ;set symbols
    : FOR L=2 ;O 15
    : GOSUB '139
    : NEXT L
-1900 REM      ;set go to ...
    : GOSUB '100("dt6g,10p,2g,21p,6p")
    : GOSUB '129
 1910 RETURN

 1920 REM %↑
```

62

```
    practice all symbols (answers)

-1930 REM      ; set heading
    : GOSUB '122
 1940 REM      ; define tabs
    : GOSUB '100("dt6g,10p,2g,21p,6p")
 1950 REM      ; set col hdgs
    : AND (Q$(2),7F)
    : GOSUB '100("bt")
    : M0=2
    : GOSUB '130(STR(Q$(2),1,24))
    : GOSUB '100("tc")
    : GOSUB '130(STR(Q$(2),26,38))
    : GOSUB '100("tc")
    : GOSUB '100("et")
    : M0=1
 1960 FOR L=3 TO 15
    : GOSUB '111(7)
    : V8=V8-7
    : GOSUB '128(26)
    : NEXT L
 1970 REM      ; set goto ref
    : GOSUB '129
 1980 RETURN

 1990 REM %↑
```

```
index

-2000 REM      ; set heading
    : GOSUB '122
    : IF STR(Q$(1),6,5) <> "INDEX" THEN 1110
 2010 REM       ;define tabs
    : GOSUB '100("dt9g,27p,9p")
 2020 REM       ;set page address col head
    : GOSUB '100("bt")
    : M0=2
    : GOSUB '130(Q$(2))
    : GOSUB '100("tr")
    : GOSUB '100("et")
    : M0=1
    : GOSUB '111(16)
    : V8=V8-16
 2030 REM       ;set chapter and pages
    : FOR L=3 TO 15
    : IF Q$(L)<>" "THEN 2040
    : GOSUB '111(16)
    : GOTO 2071

-2040 IF STR(Q$(L),10,1)=" "THEN 2060
    : IF STR(Q$(L),58,1)<>HEX(5F) THEN 2050
    : REM      ;set rule
    : GOSUB '100("bt")
    : GOSUB '102(1)
    : GOSUB '100("li")
    : GOTO 2070

-2050 REM     ;chapter head -- set chapter - insert white space - set page #
    : GOSUB '111(12)
    : V8=V8-12
    : GOSUB '100("bt")
    : IF STR(Q$(L),6,1)=" "THEN GOSUB '100("is3p")
    : GOSUB '130(STR(Q$(L),6,50))
    : GOSUB '100("ws")
    : GOSUB '99(STR(Q$(L),57,4))
    : GOTO 2070

-2060 REM      ;contents -- insert space - set contents - insert space OR set sub
      bols
    : MAT SEARCHQ$(L),="%"TO W$()
    : IF W$(1)<>HEX(0000)THEN 2061
    : GOSUB '100("du3")
    : GOSUB '100("is15p")
    : GOSUB '130(Q$(L))
    : GOSUB '100("is12p")
    : GOSUB '100("jl")
    : GOSUB '100("du2")
    : GOTO 2071

-2061 IF T5 = 5 THEN 2062
    : GOSUB '100("dt18p,3p,3p,3p,3p,3p,12p")
    : T5 = 5
-2062 GOSUB '139
    : IF L=15 THEN 2071
    : MAT SEARCH Q$(L+1),= "%" TO W$()
    : IF W$(1)<>HEX(0000) THEN 2071
    : GOSUB '100("dt9g,27p,9p")
    : T5 = 1
    : GOTO 2071

-2070 REM      ;justify line
    : GOSUB '100("tj")
    : GOSUB '100("et")
-2071 NEXT L
 2080 REM      ; set continue and goto
    : IF STR(Q$(16),41,4)="To C"THEN STR(Q$(16),40,19)="Index continued on "
    : IF STR(Q$(16),46,5)="End o"THEN Q$(16)=" "
    : IF Q$(16)<>" "THEN GOSUB '100("dt9p,30p,6p")
    : GOSUB '129
 2089 RETURN

 2090 REM %+
```

```
   subroutine (98)

2100 REM %

          add header

2110 DEFFN'98(K$)
   : L2$() = L2$() & K$ & HEX(0D)
   : GOSUB '99(" ")
   : RETURN

2120 REM %

          subroutine (99)

2130 REM %

          add string /K$/ to output

2140 DEFFN'99(K$)
2150 REM      ; add string to output buffer
   :     L2$()=L2$() & K$
2160 REM      ; print status of output buffer
2170 GOSUB '101(17,1,0,0)
   : L3$()=L2$()
   : $TRAN(L3$(),HEX(5B7B5D7D810E820F2020))R
   : $TRAN(L3$(),"................")
   : $TRAN(L3$(),HEX(40202020))R
   : SELECT PRINT 005(64)
   : PRINT HEX(06);
   : PRINT L3$()
   : PRINTUSING "current page format  [##]".VAL(STR(Q$(1).64))
2180 REM      ; if output buffer not full
   : IF L2$(5)=" " THEN 2260
2190 O$()=L2$()
2200 L2$()=L2$(5)
2210 $TRAN(O$(),X2$)R
2220 DATA SAVE BA T #1,(03,03) O$()
2230 REM      ; testing only (operator wait):     KEYIN A2$
2240 REM      ; if disk fill not full
   : IF 03 < 02-1   THEN 2260
2250     STOP "output file full ........."
2260 RETURN

2270 REM %

          subroutine (100)

2280 REM      ; add command /K$/ to output
2290 DEFFN'100(K$)
2300     L2$()=L2$() & HEX(7B) & K$ & HEX(7D)
   : GOSUB '99(" ")
2310 RETURN

2320 REM %
```

subroutine (111)

```
2330 REM      ; add extra leading (line advance)
2340 DEFFN'111(P)
2350      CONVERT P TO P$,(##)
2360      K$="el" & P$
2370 GOSUB '100(K$)
2380 RETURN

2390 REM %
```

subroutine (102)

```
2400 REM      ; invoke new mode
2410 DEFFN'102(M2)
2420 REM      ;if mode already invoked - exit
   : IF M2=M1 THEN 2470
2430      M1=M2
2440 CONVERT M1 TO M$,(#)
2450 K$ = "g" & M$
2460 GOSUB '100(K$)
2470 RETURN

2480 REM %↑
```

```
   subroutine (121)

2490 REM I

      set the next 'c' lines -or- until a blank line is incountered

2500 DEFFN'121(C)
2510 REM      : init line length
   : IF L=1 THEN L0=58
   : ELSE L0=64
2520 REM      ;find next non-blank line
   : IF L < 16 AND STR(Q$(L),,L0)<>" "  THEN 2530
   : L= L + 1
   : IF L > 16    THEN 2570
   : GOTO 2520

2530 REM      ;init line length
   : IF L=1 THEN L0=52
   : ELSE L0=64
 2540 REM      ;set line
   : GOSUB '130(STR(Q$(L),,L0))
   : GOSUB '100("qc")
 2550 REM      : decresent line count
   : C= C-1
   : IF C=0 THEN 2570
 2560 REM      ;set next line if non-blank
   : L=L +1
   : IF L< 16 AND Q$(L)<>" " THEN 2530
2570 RETURN

 2580 REM I+
```


67

subroutine (122)

```
2590 REM %
```

set line 1 (heading)

```
2600 DEFFN'122
2610 REM      ;add lead bove heading
   : GOSUB '111(99)
2620 REM      ;invoke heading mode
   : M0=3
2630 REM      ;strip underline
   : $TRAN(STR(Q$(1),6,53),X3$)7F R
2640 REM      ;single col / 1 inch margins
   : GOSUB '100("dt6q,33p,6p")
   : GOSUB '100("bt")
2650 REM      ;set hdg w/ quad lft
   : GOSUB '130(STR(Q$(1),6,53))
   : GOSUB '100("tl")
   : GOSUB '100("et")
2660 REM      ;add lead above rule
   : GOSUB '111(36)
2670 REM      ;set rule
   : GOSUB '100("bt")
   : GOSUB '100("li")
   : GOSUB '100("tj")
   : GOSUB '100("et")
2680 REM      ;lead below rule
   : GOSUB '111(45)
2690 RETURN

2700 REM %↑
```

```
    subroutine (123)

 2710 REM %
```

set current line in 2-col format

```
 2720 DEFFN'123
 2730 REM      ;check for line 16 (goto ref)
    : IF L<16 THEN 2740
    : GOSUB '129
    : GOTO 2880

-2740 REM      ;compute start of text
    : L1=POS(Q$(L)<>" ")
 2750 REM      ;not a blank line
    : IF L1> 5 THEN 2755
    : GOSUB '111(16)
    : GOTO 2880

-2755 IF STR(Q$(L),25,1)<>"."THEN 2756
    : GOSUB '111(7)
    : V8=V8-7
-2756 IF STR(Q$(L),28,1)<>"."OR STR(Q$(L).24.3)<>" "THEN 2757
    : GOSUB '111(4)
    : V8=V8-4
    : GOTO 2760

-2757 IF L1<>26THEN 2760
    : GOSUB '111(3)
    : V8=V8-3
-2760 REM      ;begin tab
    : GOSUB '100("bt")
 2770 REM      ;set sidehead (if any)
    : IF L1>23 THEN 2780
    : M0=2
    : GOSUB '130(STR(Q$(L),6,18))
-2780 REM      ;set text column
    : GOSUB '100("tl")
    : M0=1
 2790 L1=POS(STR(Q$(L),24,41)<>" ") +23
    : IF L1=27 THEN STR(Q$(L),24,3)=HEX(1B1A20)
    : IF L1=26 THEN STR(Q$(L),24,4)=HEX(18180920)
    : IF L1=28 THEN STR(Q$(L),24,4)=HEX(18181920)
    : IF L1<29 THEN L1=24
 2800 L1$()=STR(Q$(L),L1)
 2810 L2=LEN(L1$())
 2820 T$="tl"
    : IF L1=24 THEN 2850
 2840 REM      ; double indent/set
    : GOSUB '100("is6p")
    : T$="tr"
    : GOTO 2860

-2850 REM      ; not horz rule
    : IF POS(STR(Q$(L),L1,L2) <> 5F) <> 0 THEN 2860
    : GOSUB '102(1)
    : GOSUB '100("li")
    : GOSUB '100("tj")
    : GOTO 2870

-2860 REM      ; set 2nd col
    : GOSUB '130(STR(Q$(L),L1,L2))
    : GOSUB '100(T$)
-2870 REM      ; end tabs
    : GOSUB '100("et")
-2880 RETURN

 2890 REM %+
```

subroutine (124)

```
2900 REM %
```

set 2 col criterion test

```
2910 DEFFN'124
2920 REM      ; if line not blank
   : IF Q$(L) <> " " THEN 2930
   : GOSUB '111(16)
   : GOTO 3080

-2930 REM     ; begin tab
   : GOSUB '100("bt")
 2940 REM     ; set item number for col [1]
   : L1 = POS(Q$(L)<>" ")
   : IF L1 > 8 THEN 2960
 2950 GOSUB '102(1)
   : GOSUB '99(STR(Q$(L),L1,3))
-2960 GOSUB '100("tr")
 2970 REM     ; set symbol for col   [1]
   : IF STR(Q$(L),10,16)=" " THEN 2990
 2980 GOSUB '130(STR(Q$(L),10,POS(STR(Q$(L).10.17)=" ")-1))
-2990 GOSUB '100("tc")
 3000 REM     ; set rule for col [1]
   : GOSUB '102(1)
   : GOSUB '100("li")
   : GOSUB '100("tj")
 3010 REM     ; set item no. for col [2]
   : IF STR(Q$(L),33,32)=" " THEN 3080
 3020 L1=POS(STR(Q$(L),32,33)<>" ")+31
 3030 GOSUB '102(1)
   : GOSUB '99(STR(Q$(L),L1,3))
 3040 GOSUB '100("tr")
 3050 REM     ; SET SYMBOL FOR COL [2]
   : GOSUB '130(STR(Q$(L),L1+4,POS(STR(Q$(L).10.17)=" ")-1))
 3060 GOSUB '100("tc")
 3070 REM     ; SET RULE FOR COL [2]
   : GOSUB '102(1)
   : GOSUB '100("li")
   : GOSUB '100("tj")
-3080 REM     ; end tab
   : GOSUB '100("et")
 3090 RETURN

 3100 REM %↑
```

subroutine (125)

```
3110 REM %
```

set 3-col def-symbol-mem aid

```
3120 DEFFN'125
3130 REM      ; if line not blank
   : IF Q$(L)<>" " THEN 3140
   : GOSUB '111(16)
   : GOTO 3190

-3140 IF STR(Q$(L),27,15)=" "THEN 3150
   : GOSUB '111(4)
   : V8=V8-4
-3150 REM      ; begin tabular
   : GOSUB '100("bt")
   : REM      ; set col-1 /qr
   : GOSUB '130(STR(Q$(L),1,25))
   : GOSUB '100("tr")
 3160 REM      ; set 2nd col / qc
   : GOSUB '130(STR(Q$(L),27,15))
   : GOSUB '100("tc")
 3170 REM      ; set col 3 / ql
   : GOSUB '130(STR(Q$(L),43,20))
   : GOSUB '100("tl")
 3180 REM      ; end tabular
   : GOSUB '100("et")
-3190 RETURN

 3200 REM %↑
```

```
subroutine (126)

:       REM I
        set current line in 2-col CRITERION TEST format

3210 DEFFN'126
3220 REM      ;compute start of text
   : L1=POS(Q$(L)<>" ")
3230 REM      ;not a blank line
   : IF L1> 5 THEN 3235
   : GOSUB '111(16)
   : GOTO 3300

-3235 IF STR(Q$(L),23,1)<>"."THEN 3236
   : GOSUB '111(7)
   : V8=V8-7
   : GOTO 3240

-3236 IF STR(Q$(L),26,1)<>"."OR STR(Q$(L),22,3)<>" "THEN 3240
   : GOSUB '111(4)
   : V8=V8-4
-3240 REM      ;begin tab
   : GOSUB '100("bt")
   : REM      ;set sidehead (if any)
   : IF L1>21 THEN 3250
   : M0=2
   : GOSUB '130(STR(Q$(L),6,16))
 3245 REM      ;begin tab
   : GOSUB '100("bt")
   : REM      ;set sidehead (if any)
   : IF L1>21 THEN 3250
   : M0=2
   : GOSUB '130(STR(Q$(L),6,16))
-3250 REM      ;set text column
   : GOSUB '100("tl")
   : M0=1
   : L1=POS(STR(Q$(L),22,43)<>" ") +21
   : IF L1=25 THEN STR(Q$(L),22,3)=HEX(1B1A20)
   : IF L1=28 THEN STR(Q$(L),22,6)=HEX(1B1A201B1A20)
   : IF L1<29 THEN L1=22
   : L1$()=STR(Q$(L),L1)
   : L2=LEN(L1$())
   : T$="tl"
   : IF L1=22 THEN 3270
 3260 REM      ; double indent/set
   : GOSUB '100("im10p")
   : T$="tr"
   : GOTO 3280

-3270 REM      ; not horz rule
   : IF POS(STR(Q$(L),L1,L2) <> 5F) <> 0 THEN 3280
   : GOSUB '102(1)
   : GOSUB '100("li")
   : GOSUB '100("tj")
   : GOTO 3290

-3280 REM      ; set 2nd col
   : T$="tl"
   : GOSUB '130(STR(Q$(L),L1,L2))
   : GOSUB '100(T$)
-3290 REM      ; end tabs
   : GOSUB '100("et")
-3300 RETURN

 3310 REM I↑
```

subroutine (127)

```
 3320 REM %
```

set 2-col symbol / answer

```
 3330 DEFFN' 127
 3340 REM      ; if line not blank
    : IF Q$(L)<>" " THEN 3350
    : GOSUB '111(16)
    : GOTO 3480

-3350 REM      ; begin tab
    : GOSUB '100("bt")
 3360 REM      ; set item no. for col [1]
    : L1=POS(Q$(L)<>" ")
    : IF L1 > 8 THEN 3380
 3370 GOSUB '102(1)
    : GOSUB '99(STR(Q$(L),L1,3))
-3380 GOSUB '100("tr")
 3390 REM      ; set symbol for col [1]
    : IF STR(Q$(L),10,16)=" " THEN 3410
 3400 GOSUB '130(STR(Q$(L),10,POS(STR(Q$(L),10,16)=" ")-1))
-3410 GOSUB '100("tc")
 3420 REM      ; test for final criterian tests
    : IF VAL(STR(Q$(1),64)) = 39 THEN C8 = 28
    : ELSE C8=26
 3430 REM      ; tab / invoke text mode
    : L1=POS(STR(Q$(L),C8)<>" ")+C8-1
    : L1$()=STR(Q$(L),L1)
    : L2=LEN(L1$())
    : T$="tl"
    : IF L1 = C8 THEN 3460
    : IF L1 > 50 THEN 3450
 3440 REM      ; single indent
    : GOSUB '100("is3p")
    : GOTO 3460

-3450 REM      ; double indent/set
    : GOSUB '100("is12p")
    : T$="tr"
-3460 REM      ; set 2nd col
    : GOSUB '130(STR(Q$(L),L1,L2))
    : GOSUB '100(T$)
 3470 REM      ; end tabs
    : GOSUB '100("et")
-3480 RETURN

 3490 REM %+
```


73

subroutine (128)

```
3500 REM %
```

set 2-col symbol / def

```
3510 DEFFN'128(C8)
3520 REM      ; if line not blank
   : IF Q$(L)<>" " THEN 3530
   : GOSUB '111(16)
   : GOTO 3620

-3530 REM     ; begin tab
   : GOSUB '100("bt")
 3540 REM     ; set symbol  for col [1]
   : L1=POS(Q$(L)<>" ")
   : IF L1>20 THEN 3560
 3550 GOSUB '130(STR(Q$(L),1,20))
-3560 GOSUB '100("tc")
 3570 REM     ; tab / invoke text mode
   : L1=POS(STR(Q$(L),C8,38)<>" ")+C8-1
   : L1$()=STR(Q$(L),L1)
   : L2=LEN(L1$())
   : T$="t1"
   : IF L1 = C8 THEN 3600
   : IF L1 > 50 THEN 3590
 3580 REM     ; single indent
   : GOSUB '100("is3p")
   : GOTO 3600

-3590 REM     ; double indent/set
   : GOSUB '100("is12p")
   : T$="tr"
   : GOTO 3600

-3600 REM     ; set 2nd col
   : GOSUB '130(STR(Q$(L),L1,L2))
   : GOSUB '100(T$)
 3610 REM     ; end tabs
   : GOSUB '100("et")
-3620 RETURN

 3630 REM %↑
```

```
     subroutine (129)

3640 REM %

          set goto ref

3650 DEFFN'129
   : L=16
   : IF V8>0THEN 3660
   : V8=V8+V9-1
   : V9=1
3660 REM      ; compute start of text
   : L1=POS(Q$(L)<>" ")
3670 REM      ; if line not blank
   : IF L1 < 6  THEN 3740
3680 REM      ; begin tab
   : IF STR(Q$(L),L1,12)<>"To Continue" AND STR(Q$(L).L1.12)<>"To continue"THEN
     3685
   : STR(Q$(L),L1,12)=" "
   : L1=L1+12
3685 STR(Q$(L),L1+1.58-L1)=OR ALL(20)
   : GOSUB '100("bt")
3690 REM      ; tab / invoke text mode
   : GOSUB '100("tl")
   : M0=1
3700 L1=POS(STR(Q$(L),24,41)<>" ")+23
3710 L1$()=STR(Q$(L),L1)
3720 L2=LEN(L1$())
3730 T$="tl"
   : IF L1 > 35 THEN 3740
   : IF L1 > 24 THEN GOSUB '100("is4p")
   : GOSUB '130(STR(Q$(L),L1,L2))
   : GOSUB '100(T$)
   : GOSUB '100("et")
   : GOSUB '111(V8)
   : GOSUB '111(V9)
   : GOSUB '111(16)
   : GOTO 3770

3740 REM      ; set continue and goto
   : GOSUB '111(V8)
   : GOSUB '111(V9)
   : GOSUB '111(16)
   : IF Q$(L)<>" "THEN GOSUB '130(STR(Q$(L).L1.L2))
   : IF Q$(L)<>" "THEN GOSUB '100("tr")
   : ELSE GOSUB '100("el16")
3760 REM      ; end tabs
   : IF Q$(L)<>" "THEN GOSUB '100("et")
3770 V8=99
   : V9=99
   : RETURN

3780 REM %↑
```

subroutine (130)

```
 3790 REM %
```

set a line

```
 3800 DEFFN'130(L1$())
 3810 B$=HEX(7F)
 3820 REM      ; compute start of text
    : L1=POS(L1$()<>" ")
 3830 REM      ; not a blank line
    : IF L1<>0  THEN 3840
    : GOTO 3990

-3840 L1$()=STR(L1$(),L1,64-L1)
 3850 L2=LEN(L1$())
 3860 REM      ; test for all caps
    : IF POS(L1$()> 5A) <> 0  THEN 3880
    : IF POS(L1$()= "%") <> 0  THEN 3880
 3870 B$=HEX(19)
-3880 REM      ; translate space chars/build table
    : $TRAN(L1$(),HEX(20A0))R
    : MAT SEARCH L1$()<1,L2-1>, =HEX(20) TO W$()
 3890 W=0
 3900 W0=0
-3910 REM      ; delimit first / next word
    : W7=W0
 3920 W=W+1
 3930 W0=VAL(W$(W),2)
 3940 IF W0=0 THEN W0=L2+1
    : REM last word?
 3950 IF W0-W7<>1 THEN GOTO 3960
    : IF VER(STR(L1$(),W0+1,1),"#")=1 THEN 3910
    : IF VER(STR(L1$(),W0+1,1),".")=1 THEN 3910
    : IF VER(STR(L1$(),W0+1,1),")")=1 THEN 3910
    : T$="tj"
    :  GOTO 3910

    : REM IF NO WORD BETWEEN TWO SPACES - TERMINATE WITH AN EL
-3960 IF W=1 THEN 3970
    : GOSUB '99(B$)
    : REM INSERT SPACE BAND
-3970 GOSUB '140(STR(L1$(),W7+1,W0-W7-1))
    : REM SET WORD
 3980 IF W0<>L2+1 THEN 3910
-3990 RETURN

 4000 REM %↑
```

```
     subroutine (139)

 4010 REM %
          set symbol table

 4020 DEFFN'139
 4030 REM      ;if line not blank
    : IF Q$(L)<>" "THEN 4040
    : GOSUB '111(16)
    : GOTO 4070

-4040 REM      ;begin tab, set side head (if anu)
    : GOSUB '111(7)
    : V8=V8-7
    : GOSUB '100("bt")
    : M0=2
    : GOSUB '130(STR(Q$(L),6,9))
    : GOSUB '100("tl")
    : M0=1
 4050 REM      ;set symbols
    : MAT SEARCH Q$(L),="%"TO W$()
    : W=0
-4055 W=W+1
    : W0=VAL(W$(W),2)
    : IF W0=0THEN 4060
    : GOSUB '141(STR(Q$(L),W0,4))
    : GOSUB '100("tc")
    : GOTO 4055

-4060 REM      ;end tabs
    : GOSUB '100("et")
-4070 RETURN

 4080 REM %+
```

4090 REM % TAEG Report No. 88

set single word with symbol look-up

```
4100 DEFFN'140(W$)
4110 REM ? 1, 3, or 4 char word ?
   : REM WORD LENGTH =     0,    1,    2.    3.    4.   >4
   : ON W0-W7 GOTO      4120, 4130, 4210. 4140. 4160
   : GOTO  4210

-4120 STOP "0 - word length"
-4130 REM     ;1-char wd    ? ? vertical rule ?
   : IF W$<> "†" THEN 4210
   : W$="$so$u"
   : GOTO 4230

-4140 REM     ;3-char wd ... ? elipses or set # ?
   : IF VER(W$,"X##")=3 AND STR(W$,1,1)="#" THEN 4180
   : IF W$<> "..." THEN 4210
   : GOSUB '102(M0)
   : GOSUB '100("ln")
   : IF W0 <> L2+1 THEN 4150
   : IF L=16THEN 4150
   : STR(Q$(L+1),22,37)=OR ALL(20)
   : IF POS(Q$(L+1)<>" ") < 50 THEN GOSUB '100("is8p")
   : ELSE GOSUB '100("is5p")
   : T$ = "tj"
   : GOTO 4250

-4150 T$ = "tj"
   : IF VER(STR(L1$(),L2-3,4),"#A##")<>4THEN 4250
   : STR(L1$(),W0,L2-4-W0)=OR ALL(20)
   : GOTO 4250

-4160 REM ? page ?
   : IF VER(W$,"#A##")<>4 THEN 4170
   : GOSUB '102(4)
   : IF STR(W$,1,1)="1" THEN W$=STR(W$,2,3)
   : GOTO 4240

-4170 REM ? set # ?
   : IF VER(W$,"X##:")=4 AND STR(W$,1,1)="#" THEN 4180
   : ELSE GOTO 4200
-4180 REM strip leading "0" from set #
   : IF STR(W$,2,1)="0" THEN W$=STR(W$,3,2) &HEX(18)
   : ELSE W$=STR(W$,2,3) & HEX(18)
   : GOTO 4230

 4190 DEFFN'141(W$)
-4200 REM ? symbol ?
   : MAT SEARCH S1$(),=W$ TO S$ STEP 4
   : IF VAL(S$,2)=0 THEN 4210
   : S=INT((VAL(S$,2)-1)/4)+1
   : W$=S3$(S,1)
   : GOSUB '102(8)
   : GOTO 4240

-4210 REM     ; ? underlined ?
   : MAT SEARCH W$, > HEX(7F) TO S$
   : S=VAL(S$,2)
   : IF S=0 THEN 4230
 4220 REM underlined - treat char/char
   : FOR C1=1 TO W0-W7-1
   : IF STR(W$,C1,1) > HEX(7F) THEN GOSUB '102(2)
   : ELSE GOSUB '102(M0)
   : K$=STR(W$,C1,1) AND HEX(7F)
   : GOSUB '99(K$)
   : REM       ;set char
   : NEXT C1
   : GOTO  4250

-4230 REM     ;re-invoke current mode
   : GOSUB '102(M0)
-4240 REM set word
   : GOSUB '99(W$)
-4250 RETURN
```

LIST.IT

This program contains the necessary instructions to obtain a listing of the composed text created by the TYPE.TST program. It may be used for proofing the output of the TYPE.TST program or for correcting minor transmission errors should they occur.

LIST.IT

```
 0010 I## pgm: AUTHOR System 3 (typesetter transmission)
 0020 REM
    :       DIM Q2$(4)64                                  : declare i/o buffer
 0030 REM
    :       DIM Q1$(4)64                                  : declare eof buffer
    :       INIT(00)Q1$()
 0040 REM
    :       DIM Q3$(4)64                                  : declare temp i/o buffer
 1000 REM                                                 : open typesetter file
    : SELECT #1 350
    : DATA LOAD DC OPEN T#1,"TYPESET"
    : LIMITS T#1,01,02,03
 1020 SELECT PRINT 005(80)
    : PRINT HEX(0306)
    : PRINT AT(10,10);
    : INPUT "TOUCH 'RETURN(EXEC)' TO START SENDING....".R7$
    : PRINT HEX(03)
 1030 INIT (40)Q2$()
    : STR(Q2$(2),15,30)="    ARE YOU READY...    "
    : GOSUB ' 112
-1060 REM                                                 : load data record from disk
    : DATA LOAD BA T #1,(03,03) Q2$()
    : IF    Q2$() = Q1$()      THEN  1110
 1080 REM                                                 : send data to typesetter
    : GOSUB '112
    : GOTO  1060

-1110 REM                                                 : send end of transmission
    : GOSUB ' 112
    : END

 1200 REM X


 1210 REM **
 1220 REM **  Subroutine  ********************************************** 1 1 6
 1230 REM **  Abstract >
 1240 REM **    To Send typesetter data file to typesetter using its
 1250 REM **    X-on / X-off  telecommunications option.
 1290 REM **
 1300 REM ************************************************
 1310 DEFFN ' 112
 1320 REM                                                 : print current block
    : SELECT PRINT 005(64)
    : PRINT AT(15,0,);HEX(06)
    : PRINT AT(2,0);
    : Q3$()=Q2$()
    : $TRAN (Q3$(),HEX(5B0E5D0F2020))R
    : PRINT Q3$()
    : SELECT PRINT 005(80)
    : Q3$()=Q2$()
 1380 REM                                                 : dump data buffer on printer
 1390 SELECT PRINT 215(130)
    : $TRAN (Q3$(),HEX(7E7F2020))R
    : $TRAN (Q3$(),"$$$$$$$$$$$$$$$$")
 1400 FOR   J = 1 TO 4
    :       PRINT
    :       PRINT HEX(0E);Q3$(J)
    :       PRINT HEX0F(Q2$(J))
    : NEXT  J
 1410 SELECT PRINT 005(80)
 1510  RETURN

 9900 X# Program entry utility (prog.aid)
 9902 DEFFN'01"LIST SD"
 9904 DEFFN'17"      "
 9906 DEFFN'02"PRINT "
 9908 DEFFN'18"PRINTUSING "
 9910 DEFFN'03"HEX("
 9912 DEFFN'19"HEXOF("
 9914 DEFFN'04"IF    "
 9916 DEFFN'20"STR("
 9918 DEFFN'05"THEN  "
 9920 DEFFN'21"VAL("
 9922 DEFFN'06"ELSE  "
 9924 DEFFN'22"AT("
 9926 DEFFN'07"GOTO  "
```

```
 9928 DEFFN'23"ALL("
 9930 DEFFN'08"FOR   "
 9932 DEFFN'24"DEFFN ' "
 9934 DEFFN'09"STEP  "
 9936 DEFFN'25"GOSUB ' "
 9938 DEFFN'10"NEXT  "
 9940 DEFFN'26"RETURN"
 9942 DEFFN'11"ERROR "
 9944 DEFFN'27"LINPUT "
 9946 DEFFN'12"SAVE  "
 9947 DEFFN'28"RENUMBER ....-....TO....STEP.."
 9948 DEFFN'13"SCRATCH "
 9949 DEFFN'29"TRACE "
 9950 DEFFN'14"DISK  "
 9951 DEFFN'30"REM   "
 9952 DEFFN'15"SELECT "
 9953 DEFFN'31"SEARCH "
-9954 DEFFN'16
    : $GIO/005(4003)
    : $GIO/005("@P@r@o@g@r@a@s@ @A@i@d@s@ ")
    : $GIO/005("@U@t@i@l@i@t@i@t@y@ @I@n@i@t@i@a@l@i@z@a@t@i@o@n")
    : $GIO/005(400D400A400A)
 9955 $GIO/005("@s@t@a@r@t@i@n@g@ @l@i@n@e@ @n@u@a@b@e@r@:")
    : $GIO/005("@(@l@e@a@d@i@n@g@ @0@'@s@ @r@e@a@'@d@)@ ")
    : $GIO(05300639,S1$)
 9956 $GIO(710186011E16D0011E51D00171054210.S1$)
    : $GIO(710186021E26D0011E52D00171054220.S1$)
    : $GIO(710186031E36D0011E53D00171054230.S1$)
    : $GIO(710186041E46D0011E54D00171054240.S1$)
 9964 $GIO/005(400D400A)
    : $GIO/005("@i@n@c@r@:@ @(@1@-@1@0@ @u@s@e@ @0@ @f@o@r@ @1@0@)@ ")
    : $GIO(05300639710186071E76D0011E57D00171054270.S1$)
    : STRAN(S1$<7,1>,HEX(2020))0FR
    : $GIO(050015570710,S1$)
-9966 $GIO/005(400D400A)
    : $GIO/005("@a@d@d@r@e@s@:@ ")
    : $GIO(7101860B710542B071(1860C710542C07101860D710542D0400D400C.S1$)
    : IF STR(S1$,11,1)<>"3"AND STR(S1$,11.1)<>"B"THEN 9966
    : SELECT #0 <STR(S1$,11,3)>
    : ERROR GOTO 9966
 9968 $GIO/005(400D400A400A400A)
 9969 DIM S$(4)64
    : LIMITS T"stat.num",S,S1,S1
    : ERROR GOTO 9954
 9970 $GIO/005(4003400A)
    : $GIO/005("@R@E@A@D@Y@ @(@B@A@S@I@C@-@2@)")
    : $GIO/005(400A400A)
 9971 DATA LOAD BA T(S+1)S$()
    : HEXPACKSTR(S1$,9,2)FROMSTR(S1$,,4)
    : STR(S1$,9,2)=DAC STR(S1$,7,1)
    : HEXUNPACKSTR(S1$,9,2)TO STR(S$(),14.4)
    : STR(S$(),37,1)=STR(S1$,7,1)
    : DATA SAVE BA T(S+1,S1)S$()
    : LOAD DA T(S)9998,9998

 9972 DEFFN'0
    : DIM S$(4)64
    : LIMITS T"stat.num",S,S1,S1
    : ERROR GOTO 9954
 9973 DATA LOAD BA T(S+1)S$()
    : HEXPACKSTR(S1$,9,2)FROMSTR(S$(),14.4)
    : STR(S1$,9,2)=DAC STR(S$(),37,1)
    : HEXUNPACKSTR(S1$,9,2)TO STR(S$(),14.4)
    : DATA SAVE BA T(S+1,S1)S$()
    : LOAD DA T(S)9998,9998

-9998 DEFFN ' 126 "1440 "
 9999 $GIO/005(400D400C400C)
```

TRAN.TBL

This program contains the matrix of the symbol mnemonic codes and the corresponding character command strings required by the typesetter.

TRAN.TBL

```
0 $PSTAT = "CONSOLE"
1000 REM %       WEATHER SYMBOLS TABLE
1010 REM TABLE S1$() CONTAINS THE NUMERIC DESIGNATORS FOR THE CURRENT WEATHER CO
NDITION SYMBOLS.
1020 REM TABLE S3$() CONTAINS THE OUTPUT CODES REQUIRED TO CREATE THESE SYMBOLS
1030 COM S1$(128)4, S3$(128,3)13          : REM COMMON SYMBOL TABLES
1040 REM                                  : S1$() - DESIGNATOR TABLE
1050 REM                                  : S3$() - SUBSITUTION TABLE
1060 DIM B$1: B$=HEX(7B)       : REM BEGIN COMMAND ($ - for PENTA)
1070 DIM E$1: E$=HEX(7D)       : REM END COMMAND (@ - for PENTA)
1080 DIM Z$5: Z$=" ": REM WAS HEX(7B6F66357D): REM INVOKE SYMBOLS-FONT FOR 1-CHA
R
1090 %
1095 %                !                                        ! Replacement Code

1096 %   Keyword      !  Description                           !-------------------------
----!-------------------!----------------
1100 %                !                                        ! Symbol
    ! Definition              ! Memory Aid
1110 %--------------!-------------------------------!-------------------------
----!-------------------!----------------
1202 S1$( 2)="I02." : REM THUNDER                          : S3$(  2.1)=B$ & "a4" & E
$ & "T +" & B$ & "gE" & E$ & " o"
1204 S1$(  4)="WW04" : REM  SMOKE                          : S3$(  4.1)=  "W"
: S3$(  4,2)= " "      : S3$(  4,3)= " "
1205 S1$(  5)="WW05" : REM  HAZE                           : S3$(  5.1)=  "C"
: S3$(  5,2)= " "      : S3$(  5,3)= " "
1206 S1$(  6)="WW06" : REM  DUST                           : S3$(  6.1)=  "J"
: S3$(  6,2)= " "      : S3$(  6,3)= " "
1207 S1$(  7)="WW07" : REM BLOWING DUST                    : S3$(  7.1)=  "."
: S3$(  7,2)= " "      : S3$(  7,3)= " "
1208 S1$(  8)="WW08" : REM DUST WHIRLS                     : S3$(  8.1)=  "O"
: S3$(  8,2)= " "      : S3$(  8,3)= " "
1209 S1$(  9)="WW09" : REM DUST IN SIGHT                   : S3$(  9.1)=  "E"
: S3$(  9,2)= " "      : S3$(  9,3)= " "
1210 S1$( 10)="WW10" : REM TWO BAR                         : S3$( 10.1)=  "c"
: S3$( 10,2)= " "      : S3$( 10,3)= " "
1211 S1$( 11)="WW11" : REM SHALLOW FOG IN PATCHES          : S3$( 11.1)=  "j"
: S3$( 11,2)= " "      : S3$( 11,3)= " "
1212 S1$( 12)="WW12" : REM SHALLOW FOG                     : S3$( 12.1)=  "e"
: S3$( 12,2)= " "      : S3$( 12,3)= " "
1213 S1$( 13)="WW13" : REM DISTANT LIGHTNING               : S3$( 13.1)=  "o"
: S3$( 13,2)= " "      : S3$( 13,3)= " "
1214 S1$( 14)="WW14" : REM VIRGA                           : S3$( 14.1)=  "w"
: S3$( 14,2)= " "      : S3$( 14,3)= " "
1215 S1$( 15)="WW15" : REM DISTANT PRECIPATATION           : S3$( 15.1)=  "q"
: S3$( 15,2)= " "      : S3$( 15,3)= " "
1216 S1$( 16)="WW16" : REM PRECIPATATION IN SIGHT          : S3$( 16.1)=  "*"
: S3$( 16,2)= " "      : S3$( 16,3)= " "
1217 S1$( 17)="WW17" : REM DRY THUNDER                     : S3$( 17.1)=  "↑1"
: S3$( 17,2)= " "      : S3$( 17,3)= " "
1218 S1$( 18)="WW18" : REM SQUALL                          : S3$( 18.1)=  "↑5"
: S3$( 18,2)= " "      : S3$( 18,3)= " "
1219 S1$( 19)="WW19" : REM FUNNEL CLOUD                    : S3$( 19.1)=  "$s'$u"
: S3$( 19,2)= " "      : S3$( 19,3)= " "
1220 S1$( 20)="WW20" : REM DRIZZLE ENDED                   : S3$( 20.1)=  "$ss$u"
: S3$( 20,2)= " "      : S3$( 20,3)= " "
1221 S1$( 21)="WW21" : REM RAIN ENDED                      : S3$( 21.1)=  "↑0"
: S3$( 21,2)= " "      : S3$( 21,3)= " "
1222 S1$( 22)="WW22" : REM SNOW ENDED                      : S3$( 22.1)=  "$s.$u"
: S3$( 22,2)= " "      : S3$( 22,3)= " "
1223 S1$( 23)="WW23" : REM RAIN & SNOW ENDED               : S3$( 23.1)=  "$s($u"
: S3$( 23,2)= " "      : S3$( 23,3)= " "
1224 S1$( 24)="WW24" : REM FREEZING ENDED                  : S3$( 24.1)=  "K"
: S3$( 24,2)= " "      : S3$( 24,3)= " "
1225 S1$( 25)="WW25" : REM RAIN SHOWERS ENDED              : S3$( 25.1)=  "2"
: S3$( 25,2)= " "      : S3$( 25,3)= " "
1226 S1$( 26)="WW26" : REM SNOW SHOWERS ENDED              : S3$( 26.1)=  "6"
: S3$( 26,2)= " "      : S3$( 26,3)= " "
1227 S1$( 27)="WW27" : REM HAIL SHOWERS ENDED              : S3$( 27.1)=  "0"
: S3$( 27,2)= " "      : S3$( 27,3)= " "
1228 S1$( 28)="WW28" : REM FOG ENDED                       : S3$( 28.1)=  "Y"
: S3$( 28,2)= " "      : S3$( 28,3)= " "
1229 S1$( 29)="WW29" : REM THUNDER ENDED                   : S3$( 29.1)=  "M"
: S3$( 29,2)= " "      : S3$( 29,3)= " "
1230 S1$( 30)="WW30" : REM DUST DECREASED                  : S3$( 30.1)=  "D"
: S3$( 30,2)= " "      : S3$( 30,3)= " "
1231 S1$( 31)="WW31" : REM DUST, NO CHANGE                 : S3$( 31.1)=  "."
```

```
: S3$( 31,2)= " "          : S3$( 31,3)= " "
1232 S1$( 32)="WW32" : REM DUST, INCREASED                  : S3$( 32.1)=  "N"
: S3$( 32,2)= " "          : S3$( 32,3)= " "
1233 S1$( 33)="WW33" : REM HEAVY DUST, DECREASED            : S3$( 33.1)=  "T"
: S3$( 33,2)= " "          : S3$( 33,3)= " "
1234 S1$( 34)="WW34" : REM HEAVY DUST, NO CHANGE            : S3$( 34.1)=  "s"
: S3$( 34,2)= " "          : S3$( 34,3)= " "
1235 S1$( 35)="WW35" : REM HEAVY DUST, INCREASED            : S3$( 35.1)=  "d"
: S3$( 35,2)= " "          : S3$( 35,3)= " "
1236 S1$( 36)="WW36" : REM LIGHT DRIFTING. LOW              : S3$( 36.1)=  "t"
: S3$( 36,2)= " "          : S3$( 36,3)= " "
1237 S1$( 37)="WW37" : REM HEAVY BLOWING. LOW               : S3$( 37.1)=  "n"
: S3$( 37,2)= " "          : S3$( 37,3)= " "
1238 S1$( 38)="WW38" : REM LIGHT BLOWING. HIGH              : S3$( 38.1)=  "u"
: S3$( 38,2)= " "          : S3$( 38,3)= " "
1239 S1$( 39)="WW39" : REM HEAVY BLOWING. HIGH              : S3$( 39.1)=  "k"
: S3$( 39,2)= " "          : S3$( 39,3)= " "
1240 S1$( 40)="WW40" : REM FOG IN SIGHT                     : S3$( 40.1)=  "'"
: S3$( 40,2)= " "          : S3$( 40,3)= " "
1241 S1$( 41)="WW41" : REM FOG IN PATCHES                   : S3$( 41.1)=  "†2"
: S3$( 41,2)= " "          : S3$( 41,3)= " "
1242 S1$( 42)="WW42" : REM GROUND FOG, DECREASED            : S3$( 42.1)=  "†6"
: S3$( 42,2)= " "          : S3$( 42,3)= " "
1243 S1$( 43)="WW43" : REM THREE BAR, DECREASED             : S3$( 43.1)=  "?"
: S3$( 43,2)= " "          : S3$( 43,3)= " "
1244 S1$( 44)="WW44" : REM HEAVY GROUND FOG                 : S3$( 44.1)=  "†x"
: S3$( 44,2)= " "          : S3$( 44,3)= " "
1245 S1$( 45)="WW45" : REM THREE BAR                        : S3$( 45.1)=  "$sk$u"
: S3$( 45,2)= " "          : S3$( 45,3)= " "
1246 S1$( 46)="WW46" : REM GROUND FOG, INCREASED            : S3$( 46.1)=   "$sl$u"
: S3$( 46,2)= " "          : S3$( 46,3)= " "
1247 S1$( 47)="WW47" : REM THREE BAR, INCREASED             : S3$( 47.1)=   "$s)$u"
: S3$( 47,2)= " "          : S3$( 47,3)= " "
1248 S1$( 48)="WW48" : REM RIME FOG                         : S3$( 48.1)=    "X"
: S3$( 48,2)= " "          : S3$( 48,3)= " "
1249 S1$( 49)="WW49" : REM HEAVY RIME FOG                   : S3$( 49.1)=    "3"
: S3$( 49,2)= " "          : S3$( 49,3)= " "
1620 S1$( 50)="WW50" : REM ONE DRIP                         : S3$( 50.1)=      "7"
: S3$( 50,2)= " "          : S3$( 50,3)= " "
1630 S1$( 51)="WW51" : REM TWO DRIP                         : S3$( 51.1)=      "B"
: S3$( 51,2)= " "          : S3$( 51,3)= " "
1640 S1$( 52)="WW52" : REM VERTICAL TWO DRIP                : S3$( 52.1)=      "G"
: S3$( 52,2)= " "          : S3$( 52,3)= " "
1650 S1$( 53)="WW53" : REM THREE DRIP                       : S3$( 53.1)=      "R"
: S3$( 53,2)= " "          : S3$( 53,3)= " "
1660 S1$( 54)="WW54" : REM VERTICAL THREE DRIP              : S3$( 54.1)=      "L"
: S3$( 54,2)= " "          : S3$( 54,3)= " "
1670 S1$( 55)="WW55" : REM FOUR DRIP                        : S3$( 55.1)=      "S"
: S3$( 55,2)= " "          : S3$( 55,3)= " "
1680 S1$( 56)="WW56" : REM FREEZING DRIZZLE                 : S3$( 56.1)=      "A"
: S3$( 56,2)= " "          : S3$( 56,3)= " "
1690 S1$( 57)="WW57" : REM HEAVY FREEZING DRIZZLE           : S3$( 57.1)=      "b"
: S3$( 57,2)= " "          : S3$( 57,3)= " "
1700 S1$( 58)="WW58" : REM DRIZZLE & RAIN                   : S3$( 58.1)=      "r"
: S3$( 58,2)= " "          : S3$( 58,3)= " "
1710 S1$( 59)="WW59" : REM HEAVY DRIZZLE & RAIN             : S3$( 59.1)=      "l"
: S3$( 59,2)= " "          : S3$( 59,3)= " "
1720 S1$( 60)="WW60" : REM ONE DROP                         : S3$( 60.1)=      "a"
: S3$( 60,2)= " "          : S3$( 60,3)= " "
1730 S1$( 61)="WW61" : REM TWO DROP                         : S3$( 61.1)=      "s"
: S3$( 61,2)= " "          : S3$( 61,3)= " "
1740 S1$( 62)="WW62" : REM VERTICAL TWO DROP                : S3$( 62.1)=      "q"
: S3$( 62,2)= " "          : S3$( 62,3)= " "
1750 S1$( 63)="WW63" : REM THREE DROP                       : S3$( 63.1)=      "x"
: S3$( 63,2)= " "          : S3$( 63,3)= " "
1760 S1$( 64)="WW64" : REM VERTICAL THREE DROP              : S3$( 64.1)=      "-"
: S3$( 64,2)= " "          : S3$( 64,3)= " "
1770  S1$( 65)="WW65" : REM FOUR DROP                       : S3$( 65.1)=      "†3"
: S3$( 65,2)= " "          : S3$( 65,3)= " "
1780 S1$( 66)="WW66" : REM FREEZING RAIN                    : S3$( 66.1)=      "†7"
: S3$( 66,2)= " "          : S3$( 66,3)= " "
1790 S1$( 67)="WW67" : REM HEAVY FREEZING RAIN              : S3$( 67.1)=      ":"
: S3$( 67,2)= " "          : S3$( 67,3)= " "
1800 S1$( 68)="WW68" : REM RAIN & SNOW                      : S3$( 68.1)=      "("
: S3$( 68,2)= " "          : S3$( 68,3)= " "
1810 S1$( 69)="WW69" : REM HEAVY RAIN & SNOW                : S3$( 69.1)=      "$s-$u
"   : S3$( 69,2)= " "          : S3$( 69,3)= " "
1820 S1$( 70)="WW70" : REM ONE FLAKE                        : S3$( 70.1)=      "$sf$u
"   : S3$( 70,2)= " "          : S3$( 70,3)= " "
1830 S1$( 71)="WW71" : REM TWO FLAKE                        : S3$( 71.1)=      "$sa$u
```

```
*    : S3$( 71,2)= " "          : S3$( 71,3)= " "
1840 S1$( 72)="WW72" : REM VERTICAL TWO FLAKE          : S3$( 72.1)=      "Z"
     : S3$( 72,2)= " "          : S3$( 72,3)= " "
1850 S1$( 73)="WW73" : REM THREE FLAKE                 : S3$( 73.1)=      "4"
     : S3$( 73,2)= " "          : S3$( 73,3)= " "
1860 S1$( 74)="WW74" : REM VERTICAL THREE FLAKE        : S3$( 74.1)=      "8"
     : S3$( 74,2)= " "          : S3$( 74,3)= " "
1870 S1$( 75)="WW75" : REM FOUR FLAKE                  : S3$( 75.1)=      "V"
     : S3$( 75,2)= " "          : S3$( 75,3)= " "
1880 S1$( 76)="WW76" : REM ICE PRISMS                  : S3$( 76.1)=      "P"
     : S3$( 76,2)= " "          : S3$( 76,3)= " "
1890 S1$( 77)="WW77" : REM SNOW GRAINS                 : S3$( 77.1)=      "U"
     : S3$( 77,2)= " "          : S3$( 77,3)= " "
1900 S1$( 78)="WW78" : REM SNOW CRYSTALS               : S3$( 78.1)=      "F"
     : S3$( 78,2)= " "          : S3$( 78,3)= " "
1910 S1$( 79)="WW79" : REM SLEET                       : S3$( 79.1)=      "H"
     : S3$( 79,2)= " "          : S3$( 79,3)= " "
1920 S1$( 80)="WW80" : REM RAIN SHOWERS                : S3$( 80.1)=      "I"
     : S3$( 80,2)= " "          : S3$( 80,3)= " "
1930 S1$( 81)="WW81" : REM HEAVY RAIN SHOWERS          : S3$( 81.1)=      "v"
     : S3$( 81,2)= " "          : S3$( 81,3)= " "
1940 S1$( 82)="WW82" : REM VIOLANT RAIN SHOWERS        : S3$( 82.1)=      "u"
     : S3$( 82,2)= " "          : S3$( 82,3)= " "
1950 S1$( 83)="WW83" : REM RAIN & SNOW SHOWERS         : S3$( 83.1)=      "f"
     : S3$( 83,2)= " "          : S3$( 83,3)= " "
1960 S1$( 84)="WW84" : REM HEAVY RAIN & SNOW SHOWERS   : S3$( 84.1)=      "i"
     : S3$( 84,2)= " "          : S3$( 84,3)= " "
1970 S1$( 85)="WW85" : REM SNOW SHOWERS                : S3$( 85.1)=      "h"
     : S3$( 85,2)= " "          : S3$( 85,3)= " "
1980 S1$( 86)="WW86" : REM HEAVY SNOW SHOWERS          : S3$( 86.1)=      "p"
     : S3$( 86,2)= " "          : S3$( 86,3)= " "
1990 S1$( 87)="WW87" : REM SMALL HAIL SHOWERS          : S3$( 87.1)=      "z"
     : S3$( 87,2)= " "          : S3$( 87,3)= " "
2000 S1$( 88)="WW88" : REM HEAVY SMALL HAIL SHOWERS    : S3$( 88.1)=      "."
     : S3$( 88,2)= " "          : S3$( 88,3)= " "
2010 S1$( 89)="WW89" : REM HAIL SHOWERS                : S3$( 89.1)=      "↑4"
     : S3$( 89,2)= " "          : S3$( 89,3)= " "
2020 S1$( 90)="WW90" : REM HEAVY HAIL SHOWERS          : S3$( 90.1)=      "$sn$u
*    : S3$( 90,2)= " "          : S3$( 90,3)= " "
2030 S1$( 91)="WW91" : REM THUNDER ENDED. RAIN         : S3$( 91.1)=      "!"
     : S3$( 91,2)= " "          : S3$( 91,3)= " "
2040 S1$( 92)="WW92" : REM THUNDER ENDED. HEAVY RAIN   : S3$( 92.1)=      ")"
     : S3$( 92,2)= " "          : S3$( 92,3)= " "
2050 S1$( 93)="WW93" : REM THUNDER ENDED. SNOW OR HAIL : S3$( 93.1)=      "$sb$u
*    : S3$( 93,2)= " "          : S3$( 93,3)= " "
2060 S1$( 94)="WW94" : REM THUNDER ENDED. HVY SNW/HAIL : S3$( 94.1)=      "$so$u
*    : S3$( 94,2)= " "          : S3$( 94,3)= " "
2070 S1$( 95)="WW95" : REM WET THUNDER                 : S3$( 95.1)=      "$si$u
*    : S3$( 95,2)= " "          : S3$( 95,3)= " "
2080 S1$( 96)="WW96" : REM THUNDER & HAIL              : S3$( 96.1)=      "Q"
: S3$( 96,2)= " "          : S3$( 96,3)= " "
2090 S1$( 97)="WW97" : REM HEAVY WET THUNDER           : S3$( 97.1)=      "1"
   : S3$( 97,2)= " "          : S3$( 97,3)= " "
2100 S1$( 98)="WW98" : REM THUNDER & DUST              : S3$( 98.1)=      "5"
: S3$( 98,2)= " "          : S3$( 98,3)= " "
2110 S1$( 99)="WW99" : REM HEAVY THUNDER & HAIL        : S3$( 99.1)=      "9"
: S3$( 99,2)= " "          : S3$( 99,3)= " "
2115 S1$(100)="$85.": REM SNOW SHOWERS: S3$(100.1)=     "h": S3$(100.2)= " ": S3
$(100,3)=" "
2120 %--------------!-----------------------------!-----------------------
-----!----------------------!---------------
2140 FOR   I = 4 TO 99
2150       S1$    = "%"&STR(S1$(I),3,2)&"."
2160       S1$(I) = S1$
2170 NEXT  I
2190 LOAD T#0, "TYPE.TST" 0 , 6998 BEG 6999
2200 END
6999 LOAD T#0, "TYPE.INI"  7000,        BEG 0
```

TAEG Report No. 88

SEND.IT

This program contains the necessary instructions for transmitting the composed text stream created by the TYPE.TST program to the typographer.

SEND.IT

```
0000 $PSTAT = "WEST    "
0010 Z## pgm: AUTHOR System 3 (typesetter transmission)[Typographics dataphone
     number is 293-7893]
0020 REM                                        : declare i/o buffer
   :      DIM Q2$(4)64
0030 REM                                        : declare eof buffer
   :      DIM Q1$(4)64
   :      INIT(00)Q1$()
0040 REM                                        : declare temp i/o buffer
   :      DIM Q3$(4)64
0050 REM      ;tape end command - "$te"
   : DIM T$3
   : T$="$te"
1000 REM                                        : open typesetter file
   : INPUT "Enter disc containing data to be sent"D$
   : SELECT #1 (D$)
   : DATA LOAD DC OPEN T#1,"TYPESET"
   : LIMITS T#1,01,02,03
1020 SELECT PRINT 005(80)
   : PRINT HEX(0306)
   : PRINT AT(10,10);
   : PRINT "Typographics dataphone # is 293-7893"
   : INPUT "TOUCH 'RETURN(EXEC)' TO START SENDING....".R/$
   : PRINT HEX(03)
-1060 REM                                       : load data record from disk
   : $OPEN #1
   : DATA LOAD BA T #1,(03,03) Q2$()
   : IF    Q2$() = Q1$()       THEN  1110
1080 REM                                        : send data to typesetter
   : GOSUB '112
   : GOTO  1060

-1110 REM                                       : send end of transmission
   : GOSUB ' 112
   : $GIO DISK RELEASE #1 (4400)
   : DATA LOAD DC OPEN T#1,"TYPESET"
   : END

1200 REM Z


1210 REM **
1220 REM **  Subroutine  ********************************************** 1 1 2
1230 REM **  Abstract >
1240 REM **    To Send typesetter data file to typesetter using its
1250 REM **    X-on / X-off  telecommunacations option.
1290 REM **
1300 REM ***********************************************
1310 DEFFN ' 112
1320 REM                                        : print current block
   : SELECT PRINT 005(80)
   : PRINT AT(15,0,);HEX(06)
   : PRINT AT(2,0);
   : Q3$()=Q2$()
   : $TRAN (Q3$(),HEX(5B7B5D7D2020))R
   : SELECT PRINT 005(64)
   : PRINT Q3$()
   : SELECT PRINT 005(80)
1380 REM                                        : send i/o buffer to controller
   : FOR W9 = 1 TO 256
1420 REM                                        : sense for x-off char.
   : KEYIN /019,  R7$,1430,1430
   : GOTO  1460

-1430 REM                                       : not x-off char.
   :      PRINT AT(09,25,50);"# # # char recived frm tupstr (tsting for x-off)"

   :       IF STR(R7$,1,1)<>HEX(13) THEN 1460
-1440 REM                                       : wait for x-on char.
   :      KEYIN /019, R7$,1450,1450
   : PRINT AT(10,25,50);"* * * waiting for X-ON * * *"
   : KEYIN R9$,1450,1450
   : GOTO  1440

-1450 REM                                       : not x-on char.
   :     PRINT AT(08,25,50);"$ $ $ char recived frm tupstr (w/ x-on)"
   :     IF STR(R7$,1,1)<>HEX(11) THEN 1440
```

```
-1460 PRINT AT(08,25,50);
    : PRINT AT(09,25,50);
    : PRINT AT(10,25,50);
    : IF STR(Q2$(),W9,1)<>HEX(07)THEN 1465
 1461 REM release disk and delay 33 seconds
    : $CLOSE #1
    : $GIO SEND TAPE END /01D (A200, R8$) T$
    : PRINT AT(09,25,50);HEX(07);"PAUSE 33 SECONDS"
    : $GIO DELAY 33 SECONDS (010A 02FF 03FF 0400 7100 1221 4000 1913 1C14 E006)
    : PRINT HEX(07)
    : $OPEN 1462,#1
    : GOTO 1470

-1462 $GIO XMT FILLER /01D (A200,R8$) T$
    : PRINT AT(09,25,50);HEX(0707);"INPUT DISK UNAVAILABLE"
    : $OPEN 1462,#1
    : GOTO 1470

-1465 REM      ;send char to typesetter
    : $GIO SEND CHAR /01D (A200,R8$) Q2$()<W9.1>
-1470     W5 = INT((W9-1)/64)+15
    :     W6 = W9 - (W5-15)*64
    :     PRINT AT(W5,W6);STR(Q2$(),W9,1):
 1480 NEXT W9
 1500  SELECT INPUT 001
 1510  RETURN

 9900 %# Program entry utility (prog.aid)
 9902 DEFFN'01"LIST SD"
 9904 DEFFN'17"       "
 9906 DEFFN'02"PRINT "
 9908 DEFFN'18"PRINTUSING "
 9910 DEFFN'03"HEX("
 9912 DEFFN'19"HEXOF("
 9914 DEFFN'04"IF     "
 9916 DEFFN'20"STR("
 9918 DEFFN'05"THEN   "
 9920 DEFFN'21"VAL("
 9922 DEFFN'06"ELSE   "
 9924 DEFFN'22"AT("
 9926 DEFFN'07"GOTO   "
 9928 DEFFN'23"ALL("
 9930 DEFFN'08"FOR    "
 9932 DEFFN'24"DEFFN ' "
 9934 DEFFN'09"STEP   "
 9936 DEFFN'25"GOSUB ' "
 9938 DEFFN'10"NEXT   "
 9940 DEFFN'26"RETURN"
 9942 DEFFN'11"ERROR  "
 9944 DEFFN'27"LINPUT "
 9946 DEFFN'12"SAVE   "
 9947 DEFFN'28"RENUMBER ....-....TO....STEP.."
 9948 DEFFN'13"SCRATCH "
 9949 DEFFN'29"TRACE  "
 9950 DEFFN'14"DISK   "
 9951 DEFFN'30"REM    "
 9952 DEFFN'15"SELECT "
 9953 DEFFN'31"SEARCH "
-9954 DEFFN'16
    : $GIO/005(4003)
    : $GIO/005("@P@r@o@g@r@a@m@ @A@i@d@s@ ")
    : $GIO/005("@U@t@i@l@i@t@i@t@y@ @I@n@i@t@i@a@l@i@z@a@t@i@o@n")
    : $GIO/005(400D400A400A)
 9955 $GIO/005("@s@t@a@r@t@i@n@g@ @l@i@n@e@ @n@u@m@b@e@r@:")
    : $GIO/005("@(@l@e@a@d@i@n@g@ @0@'@s@ @r@e@q@'@d@)@ ")
    : $GIO(05300639,S1$)
 9956 $GIO(710186011E16D0011E51D00171054210.S1$)
    : $GIO(710186021E26D0011E52D00171054220.S1$)
    : $GIO(710186031E36D0011E53D00171054230.S1$)
    : $GIO(710186041E46D0011E54D00171054240.S1$)
 9964 $GIO/005(400D400A)
    : $GIO/005("@i@n@c@r@:@ @(@1@-@4@0@ @u@s@e@ @0@ @f@o@r@ @1@0@)@ ")
    : $GIO(0530063971018607 1E76D0011E57D0017105427 0.S1$)
    : $TRAN(S1$<7,1>,HEX(2020))0FR
    : $GIO(050015570710,S1$)
-9966 $GIO/005(400D400A)
    : $GIO/005("@a@d@d@r@e@s@s@:@ ")
    : $GIO(7101860B710542B07101860C710542C07101860D710542D0400D400C.S1$)
    : IF STR(S1$,11,1)<>"3"AND STR(S1$,11,1)<>"8"THEN 9966
    : SELECT #0 <STR(S1$,11,3)>
```

```
    : ERROR GOTO 9966
9968 $GIO/005(400D400A400A400A)
9969 DIM S$(4)64
    : LIMITS T"stat.num",S,S1,S1
    : ERROR GOTO 9954
9970 $GIO/005(4003400A)
    : $GIO/005("@R@E@A@D@Y@ @(@B@A@S@I@C@-@2@)")
    : $GIO/005(400A400A)
9971 DATA LOAD BA T(S+1)S$()
    : HEXPACKSTR(S1$,9,2)FROMSTR(S1$,,4)
    : STR(S1$,9,2)=DAC STR(S1$,7,1)
    : HEXUNPACKSTR(S1$,9,2)TO STR(S$(),14.4)
    : STR(S$(),37,1)=STR(S1$,7,1)
    : DATA SAVE BA T(S+1,S1)S$()
    : LOAD DA T(S)9998,9998

9972 DEFFN'0
    : DIM S$(4)64
    : LIMITS T"stat.num",S,S1,S1
    : ERROR GOTO 9954
9973 DATA LOAD BA T(S+1)S$()
    : HEXPACKSTR(S1$,9,2)FROMSTR(S$(),14.4)
    : STR(S1$,9,2)=DAC STR(S$(),37,1)
    : HEXUNPACKSTR(S1$,9,2)TO STR(S$(),14.4)
    : DATA SAVE BA T(S+1,S1)S$()
    : LOAD DA T(S)9998,9998

-9998 DEFFN ' 126 "1440 "
 9999 $GIO/005(400D400C400C)
```

APPENDIX B

TYPOGRAPHER'S SUBSTITUTION TABLES

TABLE 1

NTC.FD

This table was used to translate the begin tabular and end tabular command codes and the "$" code. These were required because of peculiarities of the Intercom 100 which would not allow these codes to pass on to the Penta front end system in the unmodified form.

NTC.FD

TYPE	MISC	SEARCH FOR	REPLACE WITH
01	00	^m\	$sl^m\
01	00	s^mu\	$sl^$u\
01	00	%U\	$\
01	00	%V\	#\
01	00	$tj\	/$n^\
01	00	$of]@!\	$\
01	00	$of]@s^''\	#\
01	00	%Utj\	/$n^\
01	00	^x\	$\
01	00	\	\
01	00	\	\
01	00	\	\
01	00	\	\
01	00	\	\
01	00	\	\
01	00	\	\
01	00	\	\
01	00	\	\
01	00	\	\
01	00	\	\
01	00	\	\
01	00	\	\
01	00	\	\
01	00	\	\
01	00	\	\
01	00	\	\
01	00	\	\
01	00	\	\
01	00	\	\
01	00	\	\
01	00	\	\
01	00	\	\
01	00	\	\

91

TABLE 2

NTCA.FD

This table was added to permit the use of leadering with both periods for ellipses and with em-dashes for creating horizontal rules.

NTCA.FD

TYPE	MISC	SEARCH FOR	REPLACE WITH
Ø1	ØØ	dlsl$u@\	$s$dll@$u\
Ø1	ØØ	$=\	$g\
Ø1	ØØ	$te\	$af550@OR+$te\
Ø1	ØØ	$ln@\	$dl(@$lf$s$dll@$
Ø1	ØØ	\	u\
Ø1	ØØ	^M\	^Ø\
Ø1	ØØ	$g8@^x\	$g8@$\
Ø1	ØØ	\	\
Ø1	ØØ	\	\
Ø1	ØØ	\	\
Ø1	ØØ	\	\
Ø1	ØØ	\	\
Ø1	ØØ	\	\
Ø1	ØØ	\	\
Ø1	ØØ	\	\
Ø1	ØØ	\	\
Ø1	ØØ	\	\
Ø1	ØØ	\	\
Ø1	ØØ	\	\
Ø1	ØØ	\	\
Ø1	ØØ	\	\
Ø1	ØØ	\	\
Ø1	ØØ	\	\
Ø1	ØØ	\	\
Ø1	ØØ	\	\
Ø1	ØØ	\	\
Ø1	ØØ	\	\
Ø1	ØØ	\	\
Ø1	ØØ	\	\
Ø1	ØØ	\	\

TABLE 3

DATACMA.FD

DATACMA.FD

TYPE	MISC	SEARCH FOR	REPLACE WITH
01	00	^a\	$of7@$sg$u\
01	00	^b\	$of7@3\
01	00	^c\	;cw\
01	00	^d\	$of7@$s-$u\
01	00	^e\	$of7@^5\
01	00	^f\	$of7@o\
01	00	^g\	sgu\
01	00	^h\	$of1@^3\
01	00	^i\	siu\
01	00	^j\	$of1@^4\
01	00	^k\	$of7@@\
01	00	^l\	$of7@g\
01	00	^m\	smu\
01	00	^n\	snu\
01	00	^o\	$of1@@$co$u\
01	00	^p\	spu\
01	00	^q\	$of7@g\
01	00	^r\	;rl\
01	00	^s\	ssu\
01	00	^t\	$of7@^3\
01	00	^u\	s'u\
01	00	^v\	;cl\
01	00	^w\	=\
01	00	^x\	$of1@n\
01	00	^y\	$of1@m\
01	00	^z\	$of1@)\
01	00	\	\
01	00	\	\
01	00	\	\

TABLE 4

DATACMB.FD

DATACMB.FD

TYPE	MISC	SEARCH FOR	REPLACE WITH
Ø1	ØØ	$fs\	$is\
Ø1	ØØ	$li\	$lf\
Ø1	ØØ	$tc\	*$nt\
Ø1	ØØ	$tl\	[$nt\
Ø1	ØØ	$tr\	]$nt\
Ø1	ØØ	$tw\	$dt\
Ø1	ØØ	$al\	$du2Ø\
Ø1	ØØ	$xl\	$du3Ø\
Ø1	ØØ	$co\	$^¦co\
Ø1	ØØ	^-\	$ch\
Ø1	ØØ	^	^
Ø1	ØØ	\	\
Ø1	ØØ	\	\
Ø1	ØØ	\	\
Ø1	ØØ	\	\
Ø1	ØØ	\	\
Ø1	ØØ	\	\
Ø1	ØØ	\	\
Ø1	ØØ	\	\
Ø1	ØØ	\	\
Ø1	ØØ	\	\
Ø1	ØØ	\	\
Ø1	ØØ	\	\
Ø1	ØØ	\	\
Ø1	ØØ	\	\
Ø1	ØØ	\	\
Ø1	ØØ	\	\
Ø1	ØØ	\	\
Ø1	ØØ	\	\
Ø1	ØØ	\	\
Ø1	ØØ	\	\
Ø1	ØØ	\	\

APPENDIX C

SCAT USER'S GUIDE

95

TAEG Report No. 88

SCAT USER'S GUIDE

INTRODUCTION

The SCAT system is designed specifically for use in conjunction with the computer authored texts in symbol learning created by the AUTHOR system. Performing the three functions of the user's subsystem, COMPOSITION, requires no special computer operator skills on the part of the user. The three functions are:

- COMPOSE
- LIST
- TRANSMIT.

These functions are sufficient for preparing the computer authored texts, as created by the AUTHOR system, into the form required by the typographer's subsystem, TYPESETTING.

COMPOSITION SUBSYSTEM OPERATION

INITIALIZATION. The system is initialized by clearing the CPU and loading the executive program, START. This is accomplished by performing the following steps:

1. Turn power switch ON
2. Insert SCAT SYSTEM DISK in an available disk drive
3. Key: RETURN - CLEAR - RETURN to clear the CPU
4. Key: SELECT DISK anm
   where "anm" refers to the disk drive containing the SCAT SYSTEM DISK.
   Key: RETURN
5. Key: LOAD - RUN - RETURN

This will load the executive program, START, and display the COMPOSITION subsystem function menu:

SCAT

COMPOSITION Subsystem

KEY	FUNCTION DESCRIPTION
1	COMPOSE
2	LIST
3	TRANSMIT

Depress the Function Key or number corresponding to function desired.

Depress the key corresponding to the number of the function desired. This will cause the program required to execute that function to be loaded and a display appropriate for that function will appear.

COMPOSE (Function Key 1)

This function performs the composing and replaces the mnemonic symbol codes used by the AUTHOR system with the special symbol command strings required by the typesetter. It offers the option of either composing the entire symbol learning package or composing only selected pages or frames.

The system will first inquire as to the disk address of the Programmed Instruction file created by the AUTHOR system.

COMPOSE

Enter disk address of Programmed Instruction file (---)

Insert the disk containing the Programmed Instruction file in an available disk drive and enter its disk address. The system will then initiate an inquiry into the demographic data:

COMPOSE

Enter Job Name (U.S. Navy _ _ _ _ _ _ _ _)

Enter PO Number (_ _ _ _)

Enter Date (da-mon-yr)

Enter Typographer's PO Number (_ _ _ _)

Enter Job Sequence Number; Current, First, Last

(_ _ _ _ _ _ _ _ _ _ _ _)

97

This demographic data is not critical but does provide the user and typographer with a means of identifying a particular job in the system. If the user desires, he may leave blanks for the data requested; however, leaving at least the "U.S. Navy" identifier may aid the typographer in keeping his jobs straight.

Note: Each entry of demographic data must be followed by pressing the RETURN (EXEC) key in order to be acted upon by the computer.

The system then inquires as to whether the entire programmed instruction, or only selected pages (or frames) are to be composed:

```
COMPOSE

     Enter Option from Output Selection Menu - ?

          Output Selection    1 - Compose entire text
                              2 - Compose selected pages (frames)
```

If option 1 is selected, by entering a "1" the entire package will be composed. If option 2 is selected, by entering a "2" the system will compose only those pages which are selected. In this case, it will compose each selected page and then wait for another to be selected until "STOP" is entered. When the last page (or frame) has been composed under option 1, or "STOP" has been entered under option 2, the system will add the necessary trailer information to the Composed Text file and then return to the Function Menu display.

LIST (Function Key 2)

This function prepares a listing of the Composed Text file on the lineprinter. Turn the lineprinter ON and depress the SELECT switch on the printer and the system will do the listing.

TRANSMIT

This function transmits the Composed Text file to the TYPESETTING subsystem and is, therefore, the last function of the COMPOSITION subsystem. It requires coordination with the TYPESETTING subsystem and it is, therefore, advisable to verify with the typographer that his subsystem will be ready before scheduling this function.

The system will first display a list of steps to be peformed:

TRANSMIT

1. Make sure Telecommunications Controller is connected to the Modem and set for 300 baud.
2. Turn the Modem ON.
3. Set Modem Originate/Receive switch to ORIGINATE.
4. Dial typographer's telecommunications number on telephone.
5. Place telephone handset in Modem's cradle.
6. Verify Communications Indicator Light illuminated.
7. Depress RETURN to start transmitting.

Upon performance of the listed steps in sequence, the sytem will commence transmitting the Composed Text file to the typographer in blocks of approximately 100 characters. Upon receiving STOP commands from the typographer's system, the TRANSMIT function will pause until a SEND command is received and then continue with the next block. This pause permits the TYPESETTING system to store the received data and get ready for the next block. When all of the data has been transmitted, the TRANSMIT function sends an End of Transmission command and the transmission is complete. The system may then be secured.

APPENDIX D

CODE SETS

Code Sets

Since a computer can understand numbers only and not letters and symbols, we can define each character of the alphabet as a unique number. We can further assign a unique number to any other symbol we need in our machine's language or *code set*. Using 8 bits, we can have a possible 256 unique combinations or codes (2^8). Seven bits gives a possible 128 codes (2^7) and 6 bits a possible 64 (2^6). There are several standard code sets currently in use. Some like TTS and Selectric use 6 bits, ASCII uses 7 bits, and EBCDIC uses 8 bits. The InterCom 100 internal code is a superset of ASCII and uses 8 bits. The table below shows the binary value, the decimal equivalent, octal value, hexadecimal value, and the meaning in EBCDIC, ASCII, Selectric, TTS, and InterCom 100 Internal Code.

Binary	Dec.	Octal	Hex.	EBCDIC	ASCII	Selectric Unshift	Selectric Shift	TTS	InterCom 100 Internal Code
00000000	000	000	00	NUL	NUL	Space	Space	Tape Feed	
00000001	001	001	01	SOH	SOH	!		Thin Space	• (bullet)
00000010	002	002	02	STX	STX	t	T	t	Quad Left
00000011	003	003	03	ETX	ETX	j	J	3	Quad Right
00000100	004	004	04	PF	EOT	4	$	Elevate	Quad Center
00000101	005	005	05	HT	ENQ	u	O	EM or PI	Elevate
00000110	006	006	06	LC	ACK	i	I	A	§ (section)
00000111	007	007	07	DEL	BEL		?	$	Indent (para)
00001000	008	010	08		BS	y	¢	Space Band	Em Leader
00001001	009	011	09	RLF	HT		"	Add Thin	En Dash
00001010	010	012	0A	SMM	LF	e	E	S	✓
00001011	011	013	0B	VT	VT	p	P	Em Space	¢ (cent sign)
00001100	012	014	0C	FF	FF			I	¶
00001101	013	015	0D	CR	CR			8	Return (merge)
00001110	014	016	0E	SO	SO			U	Start Command
00001111	015	017	0F	SI	SI			7	End Command
00010000	016	020	10	DLE	DLE	2	@	Return	En Leader
00010001	017	021	11	DC1	DC1	.	.	' (unquote)	⅛
00010010	018	022	12	DC2	DC2	n	N	D	¼
00010011	019	023	13	DC3	DC3		-	- (hyphen)	⅜
00010100	020	024	14	RES	DC4	r	R	R	½
00010101	021	025	15	NL	NAK			4	⅝
00010110	022	026	16	BS	SYN			I	¾
00010111	023	027	17	IL	ETB			Bell	⅞
00011000	024	030	18	CAN	CAN	o	c	N	Em Space
00011001	025	031	19	EM	EM	c	I	, (comma)	En Space
00011010	026	032	1A	CC	SUB	k	K	I	Thin Space
00011011	027	033	1B		ESC	q	Q	Quad Left	Figure Space
00011100	028	034	1C	IFS	FS	UC	UC	C	Disc. Hyphen
00011101	029	035	1D	IGS	GS	BS	BS	En Space	Hyphen
00011110	030	036	1E	IRS	RS			K	† (Dagger)
00011111	031	037	1F	IUS	US	LC	LC	QR or EM	© (copyright)
00100000	032	040	20	DS	SP	l	.	T	Space
00100001	033	041	21	SOS	!	m	M	5	! (exclamation)
00100010	034	042	22	FS	"	x	X	Z	"
00100011	035	043	23		#	g	G	) (paren)	#
00100100	036	044	24	BYP	$	0	)	L	$
00100101	037	045	25	LF	%	s	S	VR or SS	‰
00100110	038	046	26	EOB/ETB	&	h	H	W	&
00100111	039	047	27	ESC/PRE	'	\	Y	2	'
00101000	040	050	28		(	~	&	H	(

Source: Intergraphics, Inc.

Binary	Dec.	Octal	Hex.	EBCDIC	ASCII	Selectric Unshift	Selectric Shift	TTS	InterCom 100 Internal Code
00101001	041	051	29		)	r	R	Em Leader	)
00101010	042	052	2A	SM	*	d	D	Y	*
00101011	043	053	2B		+	;	:	6	+
00101100	044	054	2C		,			P	, (comma)
00101101	045	055	2D	ENQ		NI	NI	0	
00101110	046	056	2E	ACK	.	II	II	Q	. (period)
00101111	047	057	2F	BEL	/	III	III	En Leader	/
00110000	048	060	30		0	3	#	O	0
00110001	049	061	31		1	v	V	9	1
00110010	050	062	32	SYN	2	u	U	B	2
00110011	051	063	33		3	f	F	Upper Rail	3
00110100	052	064	34	PN	4	9	(	G	4
00110101	053	065	35	RS	5	w	W	;	5
00110110	054	066	36	UC	6	b	B	Shift	6
00110111	055	067	37	EOT	7	-		Lower Rail	7
00111000	056	070	38		8	8	*	M	8
00111001	057	071	39		9	a	A	. (period)	9
00111010	058	072	3A		:	c	C	X	: (colon)
00111011	059	073	3B		;	.	.	I	; (semicolon)
00111100	060	074	3C	DC4		EOT	EOT	V	
00111101	061	075	3D	NAK		II	II	Quad Cntr	
00111110	062	076	3E					Unshift	
00111111	063	077	3F	SUB	?			Delete	?
01000000	064	100	40	SP	@				@
01000001	065	101	41		A				A
01000010	066	102	42		B				B
01000011	067	103	43		C				C
01000100	068	104	44		D				D
01000101	069	105	45		E				E
01000110	070	106	46		F				F
01000111	071	107	47		G				G
01001000	072	110	48		H				H
01001001	073	111	49		I				I
01001010	074	112	4A	¢	[				[
01001011	075	113	4B	.	k				k
01001100	076	114	4C	<	l				l
01001101	077	115	4D	(	M				M
01001110	078	116	4E	+	N				N
01001111	079	117	4F	|	O				O
01010000	080	120	50	&	P				P
01010001	081	121	51		Q				Q
01010010	082	122	52		R				R
01010011	083	123	53		S				S
01010100	084	124	54		T				T
01010101	085	125	55		U				U
01010110	086	126	56		V				V
01010111	087	127	57		W				W
01011000	088	130	58		X				X
01011001	089	131	59		Y				Y
01011010	090	132	5A	!	]				]
01011011	091	133	5B	$	[				[
01011100	092	134	5C	*	\				\
01011101	093	135	5D	)	]				]

Binary	Dec.	Octal	Hex.	EBCDIC	ASCII	Selectric Unshift	Selectric Shift	TTS	InterCom 800 Internal Code
01011110	094	136	5E	;					↑
01011111	095	137	5F						
01100000	096	140	60	-					
01100001	097	141	61	/	a				a
01100010	098	142	62		b				b
01100011	099	143	63		c				c
01100100	100	144	[illegible]		d				d
01100101	101	145	65		e				e
01100110	102	146	66		f				f
01100111	103	147	67		g				g
01101000	104	150	68		h				h
01101001	105	151	69		i				i
01101010	106	152	6A	¦	j				j
01101011	107	153	6B	,	k				k
01101100	108	154	6C	%	l				l
01101101	109	155	6D	_	m				m
01101110	110	156	6E	>	n				n
01101111	111	157	6F	?	o				o
01110000	112	160	70		p				p
01110001	113	161	71		q				q
01110010	114	162	72		r				r
01110011	115	163	73		s				s
01110100	116	164	74		t				t
01110101	117	165	75		u				u
01110110	118	166	76		v				v
01110111	119	167	77		w				w
01111000	120	170	78		x				x
01111001	121	171	79	`	y				y
01111010	122	172	7A	:	z				z
01111011	123	173	7B	#	{				{
01111100	124	174	7C	@	¦				¦ (TTS Elevate)
01111101	125	175	7D	'	}				}
01111110	126	176	7E	=	~				↓
01111111	127	177	7F	"	DEL				■
10000000	128	200	80						Upper Rail
10000001	129	201	81	a					Lower Rail
10000010	130	202	82	b					Paper Feed (−)
10000011	131	203	83	c					Paper Feed (+)
10000100	132	204	84	d					
10000101	133	205	85	e					
10000110	134	206	86	f					
10000111	135	207	87	g					
10001000	136	210	88	h					
10001001	137	211	89	i					
10001010	138	212	8A						
10001011	139	213	8B						
10001100	140	214	8C						
10001101	141	215	8D						
10001110	142	216	8E						
10001111	143	217	8F						
10010000	144	220	90						
10010001	145	221	91	j					
10010010	146	222	92	k					
10010011	147	223	93	l					

Binary	Dec.	Octal	Hex.	EBCDIC	ASCII	Selectric Unshift	Selectric Shift	TTS	InterCom 100 Internal Code
10010100	148	224	94	m					
10010101	149	225	95	n					
10010110	150	226	96	o					
10010111	151	227	97	p					
10011000	152	230	98	q					
10011001	153	231	99	r					
10011010	154	232	9A						
10011011	155	233	9B						
10011100	156	234	9C						
10011101	157	235	9D						
10011110	158	236	9E						
10011111	159	237	9F						
10100000	160	240	A0						
10100001	161	241	A1						ⁿ (superior no.)
10100010	162	242	A2						'
10100011	163	243	A3	t					²
10100100	164	244	A4	u					³
10100101	165	245	A5	v					⁴
10100110	166	246	A6	w					⁵
10100111	167	247	A7	x					⁶
10101000	168	250	A8	y					⁷
10101001	169	251	A9	z					⁸
10101010	170	252	AA						⁹
10101011	171	253	AB						
10101100	172	254	AC						° (degree)
10101101	173	255	AD						′ (prime)
10101110	174	256	AE						/ (fraction bar)
10101111	175	257	AF						®
10110000	176	260	B0						©
10110001	177	261	B1						ₙ (inferior no.)
10110010	178	262	B2						₁
10110011	179	263	B3						₂
10110100	180	264	B4						₃
10110101	181	265	B5						₄
10110110	182	266	B6						₅
10110111	183	267	B7						₆
10111000	184	270	B8						₇
10111001	185	271	B9						₈
10111010	186	272	BA						₉
10111011	187	273	BB						
10111100	188	274	BC						
10111101	189	275	BD						
10111110	190	276	BE						‡
10111111	191	277	BF						
11000000	192	300	C0	{					‖
11000001	193	301	C1	A					(accent)
11000010	194	302	C2	B					(accent)
11000011	195	303	C3	C					(accent)
11000100	196	304	C4	D					(accent)
11000101	197	305	C5	E					(accent)
11000110	198	306	C6	F					(accent)
11000111	199	307	C7	G					(accent)
11001000	200	310	C8	H					(accent)
11001001	201	311	C9	I					(accent)

Binary	Dec.	Octal	Hex.	EBCDIC	ASCII	Selectric Unshift	Selectric Shift	TTS	InterCom 100 Internal Code
11001010	202	312	CA						(accent)
11001011	203	313	CB						£
11001100	204	314	CC						
11001101	205	315	CD						
11001110	206	316	CE						
11001111	207	317	CF						
11010000	208	320	D0	}					
11010001	209	321	D1	J					
11010010	210	322	D2	K					
11010011	211	323	D3	L					
11010100	212	324	D4	M					
11010101	213	325	D5	N					
11010110	214	326	D6	O					
11010111	215	327	D7	P					
11011000	216	330	D8	Q					
11011001	217	331	D9	R					
11011010	218	332	DA						¡
11011011	219	333	DB						¿
11011100	220	334	DC						ı
11011101	221	335	DD						
11011110	222	336	DE						
11011111	223	337	DF						
11100000	224	340	E0	\					& (small caps)
11100001	225	341	E1						A (small caps)
11100010	226	342	E2	S					B
11100011	227	343	E3	T					C
11100100	228	344	E4	U					D
11100101	229	345	E5	V					E
11100110	230	346	E6	W					F
11100111	231	347	E7	X					G
11101000	232	350	E8	Y					H
11101001	233	351	E9	Z					I
11101010	234	352	EA						J
11101011	235	353	EB						K
11101100	236	354	EC						L
11101101	237	355	ED						M
11101110	238	356	EE						N
11101111	239	357	EF						O
11110000	240	360	F0	0					P
11110001	241	361	F1	1					Q
11110010	242	362	F2	2					R
11110011	243	363	F3	3					S
11110100	244	364	F4	4					T
11110101	245	365	F5	5					U
11110110	246	366	F6	6					V
11110111	247	367	F7	7					W
11111000	248	370	F8	8					X
11111001	249	371	F9	9					Y
11111010	250	372	FA						Z
11111011	251	373	FB						fi
11111100	252	374	FC						fl
11111101	253	375	FD						ff
11111110	254	376	FE						ffi
11111111	255	377	FF						ffl

APPENDIX E

SAMPLE OF PROGRAMMED INSTRUCTION DEMONSTRATING THE USE OF TYPOGRAPHY

PRESENT WEATHER SYMBOLS

Symbolic Numbers

NWS-AG-A-090

November 1979

This program was developed by the
TRAINING ANALYSIS AND EVALUATION GROUP
Orlando, Florida
and the
NAVAL OCEANOGRAPHIC OFFICE
Bay St. Louis, Mississippi

TYPOGRAPHY by F. Laurence Keeler, Ph.D.

Go to 2

2

INTRODUCTION

To plot weather you must know which symbol to write for each symbolic number in the present weather position in various weather messages. This is one of several essential skills required in plotting weather information on surface charts.

The module has four parts.

In part 1 you will learn the general types of present and past weather noted in each of the ten groups on the Present Weather Chart ..... Go to 3

INTRODUCTION

In part 2 Past Weather symbol/number pairs are presented.

In part 3 you will learn the symbol/number pairs in the first 5 groups on the Present Weather Chart.

Then in part 4 you will learn the symbol/number pairs in the last 5 groups.

Learning Objective

Completion of this module will enable you to write the correct graphic symbol for each symbolic number representing the present and

4

INTRODUCTION

---

past weather in meterological reports. Given any of the 100 present weather numbers, you will be able to write the correct meaning or graphic symbol without error or hesitation, and without referring to a Present Weather or Past Weather Chart.

This lesson will help you learn easily and quickly.

For best results, follow the directions.

Go To 5

LEARNING OBJECTIVE FOR PART 1:

In this part you will learn to recall the type of present weather noted in 10 groups. Each group is a row of symbols in the Present Weather Chart. For example, you will learn that the 40s Group is for types of fog, the 50s Group is for types of drizzle, and the 80s Group is for types of showers, with past weather being the same. For example Past Weather figure 4 is for fog, figure 5 is for drizzle, etc. While there are exceptions, these rules generally are true. Being able

6

LEARNING OBJECTIVE FOR PART 1:

---

to recall which types of weather are generally noted in each of the numbered groups will make it easier for you to learn the symbolic numbers for each of the different graphic present weather symbols, and their past weather counterparts.

Go to 7

MEMORY AIDS

Practice reading the following verse until you can recall, the peg words that rhyme with each of the numbers.

Zero is for Hero
One is for Gun
Two is for Shoe
Three is for Tree
Four is for Door
Five is for Hive
Six is for Stick
Seven is for Heaven
Eight is for Gate
Nine is for Wine

Go to 8

8

MEMORY AIDS

Knowing the peg words (and sentences to be presented later which use these peg words) will help you start a chain of associations which will make it easy for you to recall the meanings of symbolic numbers.

Go to 9

MEMORY AIDS

Example of how peg words will help you remember numbers:
Given a report with the present weather number "37", note that the first digit is "3", which rhymes with the peg word "Tree". Then visualize "wind blowing through the branches of a tree." This reminds you that all present weather numbers with "3" as the first digit concern "blowing/drifting". (Later you will learn the complete definition of all present weather symbols with their respective numbers.)

116 Go to 10

10

OPTIONAL CRITERION TEST

Do you already know these groups?

Test Yourself

This criterion test is designed to test your knowledge of the present weather groups. If you already know the groups, this test will reinforce your knowledge. If you don't know the groups, it will help identify those areas to study.

Directions

1. Recall the definition of each group on the next frame.
2. Write each group number and definition on a piece of paper.

Go to 11

OPTIONAL CRITERION TEST

Group 9 _____

Group 0 _____

Group 8 _____

Group 5 _____

Group 2 _____

Group 4 _____

Group 1 _____

Group 3 _____

Group 6 _____

Group 7 _____

12

OPTIONAL CRITERION TEST

Directions

1. Check your answers now.
2. Put an X through your wrong answers.

Groups	Answers
Group 9	Thunderstorm
Group 0	Reduced visibility
Group 8	Showers
Group 5	Drizzle
Group 2	Weather ended
Group 4	Fog
Group 1	Distant weather
Group 3	Blowing/drifting
Group 6	Rain
Group 7	Snow

Go to 13

OPTIONAL CRITERION TEST

Did you pass?

1. Did you miss 2 or less of the group definitions? If so, then learn the ones that you missed by studying them on 12

   then go to 29

2. If you missed more than 2 of the group definitions, complete this learning program.

Go to 14

14

ORGANIZATION OF THE LEARNING PROGRAM

Note

Page numbers are located at the top left or right corner of each page.

Directions on how to proceed through the program are located at the bottom, right corner.

Go to 15

GROUPS TO BE LEARNED

The groups are:

Group 0	Group 1	Group 2
Group 3	Group 4	Group 5
Group 6	Group 7	Group 8
Group 9		

You will learn to recognize and define these groups in the next few pages.

Go to 16

16

LEARN GROUP NUMBERS AND DEFINITIONS

Directions

1. Look carefully on the next page at the **definition** and **memory aid** for each group.
2. Say the **memory aid** and **definition** to yourself as you look at the group.
3. Understand how the memory aid helps you remember the group.
4. Cover the **definitions**, then look at each group number and recall the **definitions**.
5. Repeat this 4 or 5 times for each of the groups.

LEARN GROUP NUMBERS AND DEFINITIONS

Definition	Group	Memory Aid
Reduced visibility	Group 0	Hero - Almost blinded by smoke(REDUCED VISIBILITY), he saved the child from the burning house.
Distant weather	Group 1	Gun - I have gun, keep your DISTANCE.
Weather ended	Group 2	Shoe - A kick in the pants ENDED the argument.

Go to 18

18

LEARN GROUP NUMBER AND DEFINITION

Definition	Group	Memory Aid
Blowing/drifting	Group 3	Tree - wind BLOWING through the branches of a TREE.
Fog	Group 4	Door - It's hard to find your DOOR in the FOG.
Drizzle	Group 5	Hive - DRIPS of honey from a hive look like the symbol for drizzle.
Rain	Group 6	Stick - a stick floats in a RAIN puddle.

Go to 19

LEARN GROUP NUMBER AND DEFINITION

Definition	Group	Memory Aid
Snow	Group 7	Heaven - Snow FLAKES float down from heaven.
Showers	Group 8	Gate - rain SHOWERS, open flood gate.
Thunderstorm	Group 9	Wine - The drunk hid in the wine cellar durng the THUNDERSTORM.

Go to 20

20

PRACTICE DEFINING THE GROUPS

Directions

1. Read all directions before you practice.
2. Try to recall the definition of each group in the exercise on page 21 .
3. **Write** your first impression of the group's definition.
4. If the definition is difficult to remember, recall the memory aid first, then recall the definition.
5. Check your answer immediately in the answer section below the practice exercise.

Go to 21

PRACTICE DEFINING THE GROUPS

Practice	Group 9	Group 4	Group 8
Exercise	Group 5	Group 3	Group 8
	Group 3	Group 4	Group 9
	Group 5	Group 9	Group 3
	Group 4	Group 8	Group 5

Group	**Definition**
Group 4	Fog
Group 9	Thunderstorm
Group 8	Showers
Group 5	Drizzle
Group 3	Blowing/drifting

Go to 22

22

PRACTACE DEFINING THE GROUPS

Practice	Group 2	Group 6	Group 0
Exercise	Group 1	Group 7	Group 1
	Group 0	Group 2	Group 6
	Group 7	Group 7	Group 0
	Group 2	Group 1	Group 6

Group	**Definition**
Group 2	Weather ended
Group 0	Reduced visibility
Group 1	Distant weather
Group 6	Rain
Group 7	Snow

Go to 23

PRACTICE DEFINING THE GROUPS

Directions

1. Keep practicing until you can define all the groups without pausing.
2. Practice defining the most difficult groups for you more than the easier groups.
3. Vary the way you go through the practice groups. (Left to right, then right to left.)
4. Keep practicing until you recall the answers without hesitating.

To practice ..... Go back to 21

After practicing ..... Go to 24

24

TEST YOURSELF

Directions

1. Copy the group number and then write the definition for each.

2. If you want a memory aid ..... go to 25

3. Refer to the memory aids only when you can't think of the definition.

Self Test

Group 3	Group 0
Group 2	Group 5
Group 1	Group 6
Group 8	Group 7
Group 4	Group 9

For Answers Go to 27

TEST YOURSELF

Groups	Memory Aids
Group 3	Tree - wind BLOWING through the branches of a TREE.
Group 2	Shoe - A kick in the pants ENDED the argument.
Group 1	Gun - I have gun, keep your DISTANCE.
Group 8	Gate - rain SHOWERS, open flood gate.
Group 4	Door - It's hard to find your DOOR in the FOG.
Group 0	Hero - Almost blinded by smoke (REDUCED VISIBILITY), he saved the child from the burning house.

TEST YOURSELF

Groups	Memory Aids
Group 5	Hive - DRIPS of honey from a hive look like the symbol for drizzle.
Group 6	Stick - a stick floats in a RAIN puddle.
Group 7	Heaven - Snow FLAKES float dow from heaven.
Group 9	Wine - The drunk hid in the wine cellar during the THUNDERSTORM.

Go back to 24

TEST YOURSELF

Group	Answers
Group 3	Blowing/drifting
Group 2	Weather ended
Group 1	Distant weather
Group 8	Showers
Group 4	Fog
Group 0	Reduced visibility
Group 5	Drizzle
Group 6	Rain
Group 7	Snow
Group 9	Thunderstorm

28

TEST YOURSELF

End of Test Directions

1. If you missed any definitions, you need more practice
   - skip groups you already know
   - spend extra time on those groups you find difficult to remember.
   - do the self test after you practice each time

   To practice .....
   go back to 21

2. If you correctly defined all the groups .... CONGRATULATIONS!

For next part of module ..... go to 29

INTRODUCTION

In part 2 of this module you will learn to recall the symbol/number pairs for Past Weather. There are 7 past weather symbols. They represent blowing/drifting phenomena, fog, drizzle, rain, snow, showers and thunderstorms.

Learning Objective Part 2

After you have finished with this part of the module you will be able to write the correct symbol for any Past Weather number.

Go to 30

30

OPTIONAL CRITERION TEST

Do you already know these symbols?

Test Yourself

1. Recall the graphic symbols for each number on the next frames.
2. Write each number and its graphic symbol on a piece of paper.

Go to 31

OPTIONAL CRITERION TEST: SYMBOLS

3 _____

4 _____

6 _____

9 _____

5 _____

7 _____

8 _____

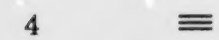
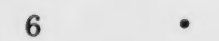
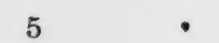
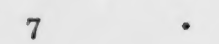
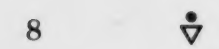
Go to 32

32

OPTIONAL CRITERION TEST

Directions

1. Check your answers now.
2. Put an X through your wrong answers.

Numbers	Answers
3	
4	
6	
9	
5	
7	
8	

Go to 33

OPTIONAL CRITERION TEST

Did you pass?

1. Did you miss 1 of these symbols? If so, then learn the one that you missed by studying it on 0032.

2. If you missed more than 1 of the symbols, complete this learning program.

Go to 34

34

ORGANIZATION OF THE LEARNING PROGRAM

Overview

The symbols will be presented in one practice group. You will learn the symbols in this group.

Go to 35

35

SYMBOL SET 1

The numbers in this set are:

3 4 5 6 7 8 9

You will learn to recognize and coordinate these numbers with their respective symbols in the next few pages.

Go to 36

36

SYMBOL SET 1: LEARN THE SYMBOLS

Directions

1. Look carefully on the next page at each **graphic symbol** and **memory aid** for each number.
2. Recall the **memory aid** and **graphic symbol** as you look at the **number**.
3. Understand how the **memory aid** helps you coordinate the **symbol** with a **number**.
4. Cover the symbols, then look at each **number** and recall the **graphic symbol**.
5. Repeat this 4 or 5 times for each of the numbers.

SYMBOL SET 1: LEARN THE SYMBOLS

Graphic Symbol	Number	Memory Aid
	3	Blowing/drifting
	4	Fog
	5	Drizzle
	6	Rain
	7	Snow
	8	Showers
	9	Thunderstorms

38

SYMBOL SET 1: PRACTICE

Directions

1. Read all directions before you practice.
2. Try to recall the graphic symbol for each number in the exercise on page 39 .
3. **Write** your first impression of the graphic symbol.
4. If the graphic symbol is difficult to remember, recall the memory aid first, then recall the graphic symbol.
5. Check your answer immediately in the answer section below the practice numbers.

Go to 39

SYMBOL SET 1: PRACTICE

Practice Numbers	4	7	6	3	5	8	9	6	7
	5	9	4	3	7	8	9	6	5
	7	5	3	4	8	3	5	7	9
	8	6	4	9	8	3	7	6	5
	5	4	9	8	5	9	7	3	6

Number	Graphic Symbol
3	
5	
6	
7	
4	
8	
9	

40

SYMBOL SET 1: PRACTICE

Directions

1. Keep practicing until you can recall the symbols for each number without pausing.
2. Practice the most difficult numbers more than the easier ones.
3. Vary the way you go through the practice numbers. (Left to right, then right to left.)
4. Keep practicing until you recall the answers without hesitating.

To practice ..... Go back to 39

After practicing ..... Go to 41

SYMBOL SET 1: TEST YOURSELF

Directions

1. **Write** the **graphic symbol** for each number in the self test. Use scratch paper.
2. If you want a memory aid ..... go to 42
3. Refer to the memory aids only when you can't think of the symbol.

Self Test

4	3
5	6
8	7
9	

For Answers Go to 43

42

SYMBOL SET 1: TEST YOURSELF

Numbers	Memory Aids
4	Fog
5	Drizzle
8	Showers
9	Thunderstorms
3	Blowing/drifting
6	Rain
7	Snow

Go Back to 41

SYMBOL SET 1: TEST YOURSELF

Numbers	Answers
4	
5	
8	
9	
3	
6	
7	

SYMBOL SET 1: TEST YOURSELF

End of Test Directions

1. If you missed any answers, you need more practice.
   - Skip symbols you already know
   - Spend extra time on those symbols/numbers you find difficult to remember.
   - do the self test after you practice each time

   To practice .....Go Back to 39

2. If you correctly coordinated all the symbols with numbers

   congratulations!

TAEG Report No. 88

DISTRIBUTION LIST

Air Force

Headquarters, U.S. Air Force (Mr. Stiegman)
Headquarters, Air Training Command (XPTD) (Dr. Schufletowski)
Headquarters, Air Training Command (XPTIA, Mr. Goldman)
Air Force Human Resources Laboratory, Brooks Air Force Base
Air Force Human Resources Laboratory (Library), Lowry Air Force Base
Air Force Office of Scientific Research/AR (Dr. A. R. Fregly)
Headquarters Tactical Air Command (DOOS) Langley Air Force Base
AFMTC/XR (Capt. Englebretson) Lackland Air Force Base
Headquarters 34 TATG/TTD (Lt. Col. Lee), Little Rock Air Force Base
Headquarters MAC/DOTF (Capt. Orler), Scott Air Force Base
Air Force Human Resources Laboratory (Mr. Johnson), Wright-Patterson Air Force Base
HQ AFLC/LOLMP, Wright-Patterson Air Force Base

Army

Commandant, TRADOC (Technical Library)
ARI (Dr. Ralph R. Canter, 316C; Dr. Edgar Johnson; Mr. James Baker; Dr. H. F. O'Neil, Jr.; Dr. Beatrice Farr, PERI-OK)
ARI Field Unit - Fort Leavenworth
ARI (Reference Service)
ARI Field Unit - Fort Knox (PERI-IK)
COM USA Armament Materiel Readiness Command (DRSAR-MAS)
COM USA DARCOM Materiel Readiness Support Activity (DRXMD-MP)
ATSC-DS-SPAS (Mr. Klesch)

Coast Guard

Commandant, U.S. Coast Guard Headquarters (G-P-1/62; G-RT/81)

Marine Corps

CMC (OT, LMD-1 (Mr. Maragides))
CGMCDEC (Mr. Greenup)
Director, Marine Corps Institute
CO MARCORCOMMELECSCOL (ITS) (Mr. D. E. Mueller)

Navy

OASN (R&D, MRA&L, Dr. Sicilia)
CNO (OP-115, M. Malehorn; OP-987H, Dr. R. Smith; OP-987, Mr. H. Stone; OP-401E, CDR Linehan)
NAVCOMPT (NCD-7)
ONR (458 (2 copies), 455)

DISTRIBUTION LIST (continued)

CNM (MAT-08T2, Mr. A. L. Rubinstein; MAT-04213, Mr. Weyburn)
CNET (01, 00A, N-5 (6 copies))
CNAVRES (02)
COMNAVSEASYSCOM (05L1C, 05L1C2, 05L3 (Mr. Shihda))
COMNAVAIRSYSCOM (03, 340F, 413G, 340C (Mr. Hurst))
CO NAVMEDRSCHDEVCOM (CAPT H. J. Connery)
CNTECHTRA (017, Dr. Kerr (5 copies))
CNATRA (Library)
COMTRALANT
COMTRALANT (Educational Advisor)
COMNAVELEXSYSCOM (Code 4505, Mr. Sibole)
NPPSMO (Code NPPO, Mr. Cherny; Code 10, Mr. Burby)
NAVSHIPWPNSYSENGSTA (Code 5700, Mr. Radcliff)
COMOPTEVFOR (Code 32, CDR Brown)
COMTRAPAC (2 copies)
CO NAVPERSRANDCEN (Library (4 copies), Code 260 (Dr. Blanchard))
NAVPERSRANDCEN Liaison (01H)
Superintendent NAVPGSCOL (2124)
Superintendent Naval Academy Annapolis (Chairman, Behavioral Science Dept.; Library)
CO NAVEDTRAPRODEVCEN (AH3; EAT, Dr. Smith; Technical Library (2 copies))
CO NAVEDTRASUPPCEN NORVA (00(2 copies); N1111, Joe Fazio)
CO NAVEDTRASUPPCENPAC (5 copies)
CO NAVAEROMEDRSCHLAB (Chief Aviation Psych. Div.)
CO FLTCOMDIRSYSTRACENPAC
CO NAMTRAGRU
CISO, NTTC Corry Station
CO NAVTRAEQUIPCEN (N-424 (2 copies), N-211, N-001, N-002)
Center for Naval Analyses (2 copies)
U.S. Naval Institute (CDR Bowler)
OIC NODAC (2)
CO TRITRAFAC (2 copies)
CO NAVSUBTRACENPAC (2 copies)
CO FLEASWTRACENPAC
CO FLETRACEN SDIEGO
CISO, SSC
Executive Director NAVINSTPRODEVDET
CO NAVTECHTRACEN Corry Station (Cryptologic Training Department)
Supply Schools Training Officer (Code 730), Meridian
Office of Civilian Personnel, Southern Field Division (Jim Herndon)
VT-10 (Education Specialist)
CO NAVSUBSCOL NLON (Code 0110)
CO NAVTECHTRACEN Treasure Island (Technical Library)
NAVAIRTECHSERVFAC (Code 04A4, Mr. Richardson; Code 00, CDR Kogler;
Code 02, Mr. Muller)
President, Naval War College (Library)

DISTRIBUTION LIST (continued)

Other

Military Assistant for Human Resources, OUSDR&E, Room 3D129, Pentagon
(CDR Paul Chatelier)
OASD (MRA&L) (LT COL Grossel)
Program Manager, Office of Cybernetics Technology, Defense Advanced Research Projects Agency (Dr. J. Dexter Fletcher)
Institute for Defense Analyses (Dr. Jesse Orlansky)
COM National Cryptologic School (Code E-2)
Hughes Aircraft Company (Mr. Bean)
Systems Exploration, Inc. (Mr. Booher)
Bio Technology, Inc. (Mr. Post)
EG&G Hydrospace-Challenger (Mr. Grubb)
Telcom Systems, Inc. (Mr. Geyer)

Information Exchanges

DDC (12 copies)
DLSIE (Mr. James Dowling)
Executive Editor, Psychological Abstracts, American Psychological Association
ERIC Processing and Reference Facility, Bethesda, MD (2 copies)